NASA-AISRP FINAL REPORT
Report Period: 10/1/2009-9/30/2010
Grant No: NNG05GA30G

Estimating Missing Data in Sensor Network Databases Using Data Mining to Support Space Data Analysis

Submitted to

Dr. Nand Lal
NASA
Phone: (301) 286-7350
Email: Nand.Lal@nasa.gov

By

Le Gruenwald
University of Oklahoma
School of Computer Science
110 W. Boyd, Room 150 DEH
Norman, OK 73019
Phone: 405-623-8358
Fax: 405-325-4044
Email: ggruenwald@ou.edu

TABLE OF CONTENTS

1. PROJECT ACCOMPLISHMENTS

In the past year (2009-2010), we were able to complete the following tasks:

- Completed the development of a general framework for mining and estimating missing sensor data and discovering knowledge for different types of sensor network.
- Completed the development of MASTER-M (Mining Autonomously Spatio-Temporal Environmental Rules for Multi-hop sensor networks), an algorithm to estimate missing sensor data and discovering knowledge in multi-hop sensor networks.
- Implemented MASTER-M using C++ and conducted experiments comparing MASTER-M with three existing estimation algorithms for data streams, SPIRIT [Papadimitriou, 2005], Average and TinyDB [Madden, 2005], using sensor data gathered from the Intel Berkley Lab sensor network [Intel, 2009] and synthetic datasets.
- Investigated additional satellite applications from NASA and gathered additional satellite datasets for further testing for MASTER-M.
- Completed the development of DEMS (A Data Mining Based Technique to Handle Missing Data in Mobile Sensor Network Applications), an algorithm to estimate missing sensor data and discovering knowledge in mobile sensor networks.
- Implemented DEMS using C++ and conducted experiments comparing DEMS with three existing estimation algorithms for data streams, SPIRIT, Average and TinyDB, using sensor data gathered from the DAPPLE project [Dapple, 2004] and synthetic datasets.
- Simulated multiple server sensor networks and extended our general framework for multiple server sensor networks, and compared the performance of our framework with the three existing techniques, SPIRIT, Average and TinyDB.
- Obtained the spectral dataset from Dr. Nikunj C. Oza, our collaborator at NASA Ames Research Center, and performed experiments using the spectral dataset to compare our algorithms with the existing algorithms, SPIRIT, Average and TinyDB.
- Provided a theoretical analysis in terms of space and time complexity for MASTER trees.
- Estimated a theoretical energy savings for data estimation over retransmission.
- Graduated one Master's student and prepared one PhD students for his PhD general exam.
- Published one conference paper, two workshop papers, and one Master's thesis; submitted one conference paper and prepared three journal papers for publication submission (eighteen publications to date).

In the following sections, we provide the details of the above tasks.

1.1. Completed the development of a general framework for mining and estimating missing sensor data and discover knowledge for different types of sensor network.

Two basic components of our technique is (1) association rule mining and (2) data estimation from association rules. We store the summary of the sensor readings and mine the association rules from the summary data. The obtained association rules are further used to estimate missing sensor readings. Each component is described in this section individually.

1.1.1. Association Rule

Association rule mining [Aggrawal, 1993] is a popular data mining strategy for transactional data. Association rule mining technique can identify the rules among the items. Consider a set of discrete items $I = \{i_1, i_2, \ldots, i_n\}$. Any subset of I of cardinality k is referred to as a k-itemset. Further, a k-itemset is said to be frequent if and only if the probability of joint occurrence of all the items in the itemset is greater or equal to a user defined minimum support. Now, let X and Y be two different items, we say that $i_p \rightarrow i_q$ is an association rule if and only if the joint probability (named as the support of the rule) of i_p and i_q is at least equal to the minimum support and the conditional probability (named as the confidence of the rule) of i_q given i_p is at least equal to the minimum confidence. i_p is referred to as the antecedent part and i_q as the consequent. Additionally, we can have a time clause associated with any association rule indicating the time period during which probabilities are to be evaluated.

Sensor network data are clearly not transactional; rather they are continuous and multi-dimensional. In order to define probability events for sensors, their data must be cast in terms of Boolean propositions. We consider the complete vector space where the reading of any sensor node may fall and divide the complete vector space into equally spaced small subspaces. In our approach each subspace is considered as a Boolean item. Mathematically this can be denoted by $S_i[a, b]$ where $[a, b]$ is the subspace and S_i is the sensor. Now as each round is composed of such items and each item is coming from each sensor, rules among the items are essentially the rules among the sensor readings. If $S_A[20, 30] \rightarrow S_B[15, 25]$ is a rule between the items $S_A[20, 30]$ and $S_B[15, 25]$ then it is a rule between the sensor readings from sensor A and sensor B. If sensors A and B are sensing temperature, the above rule reads that if A reports a temperature in the range of the 20-30 degrees Celsius then B is likely to report a temperature between 15 and 25 Celsius. Of course, we could have much more intricate rules involving any number of nodes and arbitrary data spans. In our framework we mine these kinds of rules and the rules are further used for missing data approximation. The next section introduces our data structure to mine association rules.

1.1.2. Data Structure

Designing a compact data structure for infinitely large data streams is a challenging problem. Moreover the data structure has to store sufficient information to mine arbitrary association rules among the sensor nodes; and at the same time it has to be compact and the algorithm has to be one pass. To meet all the requirements simultaneously, we propose a complex tree structure called MASTER-tree. In this section we describe our novel data structure with sufficient background details. Our data structure (MASTER-tree) shall contain sufficient data summary to be able to evaluate potential association rules as defined in the previous section. Our data

structure will satisfy the incremental and compact properties required for data streams wherein raw data are scanned once at their arrival.

1.1.3. Pattern-tree

The Pattern-tree is a graphical representation of all possible transactional itemsets proposed in [Giannella, 2003]. An example of a Pattern-tree over the set $S = \{S_1, S_2, S_3\}$ is given in Figure 1. The best way to define the Pattern-tree begins by observing that the set of all possible itemsets is exactly the power set of the transaction set, which has a cardinality of 2^n if the transaction set contains n items. Hence, there exists a bijective map between the set of all itemsets and the binary hypercube of dimension n. The Pattern-tree is in fact nothing but a spanning tree of the corresponding hypercube. The nodes on the path from the root of the Pattern-tree to any of its nodes are mapped to a unique itemset.

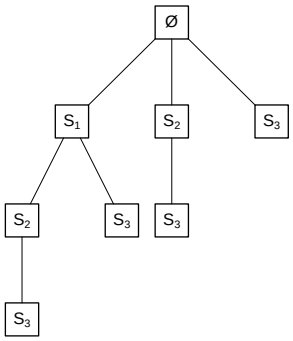

Figure 1. Pattern-tree Example of Order 3

However, the Pattern-tree does not facilitate the storing of sensor readings that are quantized into vector spaces. To overcome that short coming we proposed a Grid-based Structure (GS) with Pattern-tree that is discussed in the next section.

1.1.4. Grid-based Structure (GS)

In [Giannella, 2003], it was sufficient to store data counts at the tree node level to be able to evaluate association rule parameters (minimum support and minimum confidence). In our extended paradigm, tree nodes are not discrete items. For our purposes, we map each tree node with a set of multi-dimensional grids each being a discretization of the bounded vector space. Data counters are associated with each grid cell. As data are sampled in each round, the appropriate grid cell counters are incremented. Note that cell memory allocation can be made adaptively i.e., it is only done following the first sample falling within its boundaries. To this end, let us define the set of grids associated with each Pattern-tree node. The nodes at the first tree level correspond to singleton itemsets (their parent root being the empty set). For each of those nodes, there corresponds one single grid allowing the calculation of singleton nodesets. For each node at the second level, a GS blueprint is issued out of each cell of the one GS of its parent node. The blueprint terminology is carefully chosen to indicate that such GSs are not to be allocated unless the parent cell is, and in turn, the cells in any of the child GSs are allocated adaptively. This construction continues recursively until the last level of the tree. While conceptually the total number of cells in the entire tree grows exponentially in terms of the tree depth and the discretization granularity, one should keep in mind the notion of adaptive allocation which in practice should greatly reduce the complexity.

1.1.5. MASTER-tree

We made a couple of more amendments along with the Grid structure to Pattern-tree to meet our requirements. We would like our estimate to have a continuous value. Up to this point, association rules only tell us data ranges. Hence, if an association rule were to imply a missing sensor, we can only infer its expected range as dictated by the discretization. To define an expected value, we will additionally store information about the sample moments of the in-cell distribution, which can be represented by power sums. Note that a power sum can be updated incrementally as the streams are arriving. Also, a cell once allocated will have the same consumption no matter the stream length, hence compaction. The summary statistics vector in each cell has the form of $\langle count, \{\sum_{s \in samples} x_{i,j,s}^r \,|\, i = [1,d], j = [1,n], r = [1,m]\}\rangle$ where $x_{i,j,s}^r$ is the i-th attribute of the data point from the s-th sample reported by sensor j and raised to the r-th power. Another benefit that will be derived from the in-cell distribution information is that we can infer data counts over partial cells (by using a distribution approximation system (e.g. Pearson System [Elderton, 1969]) and numerical integration (e.g. Adaptive Simpson's Rule [McKeeman, 1962])). However, we cannot infer posterior distributions conditioned on arbitrary (i.e., covering partial cells) subspaces. Hence, the Pattern-tree model in Figure 1 only allows S_3 to have arbitrary consequent subspace given any combination of antecedent nodes. However we like to imply every node over an arbitrary consequent subspace from any combination of antecedent nodes which is not possible in the current Pattern-tree. Consider an automorphic tree to that in Figure 1 which has S_2 as the consequence node. S_2 is no longer problematic if the two automorphic graphs are unified. Similarly, we construct an automorphic Pattern-tree for every node. We call the union graph a MASTER-Tree (including the embedded GS structures). The automorphisms should be chosen such that the path from the root to every leaf contains nodes in ascending spatial distance to the leaf. This will ensure that the estimation algorithm follows an efficient tree descent when generating new rules. Figure 2 shows a MASTER-tree construction example for a spatial situation where S_2 is closest to S_3, S_3 closest to S_2, and S_1 closest to S_2.

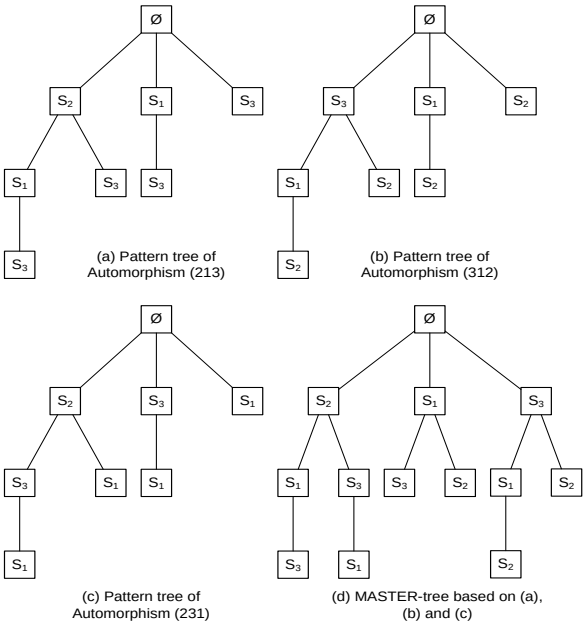

Figure 2. MASTER-tree construction example

1.1.6. Temporal Snapshots

In this work, we postulate that sensornet environments exhibit cyclic (i.e., periodic) trends. Hence, we propose to identify all expected elemental cycles. We refer to such set as time-generative basis. Any time expression is therefore a union composition of two or more periods in the time basis. To capture these temporal effects, we store a snapshot of MASTER-tree over each basic period (storing data from only the corresponding period). This allows us to generate data over any desired time period by additively combining tree data over basic periods.

1.1.7. Estimating Missing Data

The task of the estimation procedure is to autonomously and efficiently explore the rule space to (1) determine the relevant time period over which data shall be considered for rule evaluation, (2) determine the set of sensor nodes and their respective subspaces that constitute the rule (where the consequent node is the missing node), and (3) compute the estimate of the missing node as its expected value over its consequent subspace. The computed rule is evidently more interesting if its consequent missing node subspace has a "small span" as it would in turn suppress the variance of the estimate. The search over the rule space needs to be properly orchestrated so as our estimation procedure is both effective and efficient. We propose an iterative estimation method in which the estimation is adjusted progressively. The fine-tuning of the estimation can be carried on until the user-set error margin is met or until the estimation execution time is up. This way anytime the control process decides to time out the mining (when the user time bound is reached), we will always have some "ready-to-go" estimate. Since a data round may have several missing nodes each having several missing attributes, we shall run different estimation threads, each estimating one particular missing attribute of one particular missing node. That way we guarantee that the estimation time is fairly allocated amongst all estimations while each estimation thread progressively fine-tunes its estimate.

The algorithm starts by identifying the most relevant temporal period for the current estimation problem. This is fixed to the elemental period that contains the current data time stamp. The algorithm then obtains the prior distribution of the missing attribute to be estimated from the MASTER-tree. The algorithm then attempts to contain the stretch of such distribution by ignoring data in the two end margins while satisfying the support and confidence thresholds. This rule can be viewed as $\emptyset \rightarrow M$ where M is the missing node (i.e. nothing implies M). If the last step fails to satisfactorily constrain the span of M then relevant information from other streams needs to be acquired to refine the distribution of M. Meanwhile an estimate can be backed up from the rule just obtained, and in such case, the consequent subspace of M has the span higher than the allowed minimum span threshold (error bound). In reference to this parameter relaxation, such rule will be referred to as a relaxed rule. The algorithm chooses one new antecedent node to imply the posterior distribution of M's missing attribute. The node closest to the missing node is chosen as the new additional antecedent node. Such new node can be fetched in a constant time from our tree model. The initial relevant subspace for the antecedent node is chosen as the one that contains the current reading reported by the added node. If enough support cannot be found, the relevant subspace is augmented iteratively (cell by cell) until the support condition is met. The cell, the centroid of which is close to the current reading of the new node, is added in every iteration. After assuring enough rule support, the same principle of trying to constrain the posterior distribution of the missing attribute is applied.

The new support and confidence can be incrementally updated with every change of the relevant or consequent subspaces.

The integration of a new antecedent node is repeated until the estimation procedure reaches one of the three possible conditions: (1) a rule meeting the minimum support, confidence, and a consequent subspace span is found, (2) the mining process is timed-out and a relaxed rule is found, or (3) no more node is added to the prior node set (antecedent rule part) and a relaxed rule is obtained. The procedure then returns the estimate value as the final expected value computed over the consequent subspace.

1.2. Completed the development of MASTER-M (Mining Autonomously Spatio-Temporal Environmental Rules for multi-hop sensor networks), an algorithm to estimate missing sensor data and discovering knowledge in multi-hop sensor networks.

Inter-sensor communication is usually restricted to short distance due to energy and bandwidth burden [Al-Karaki, 2004]. Thus for a bigger area coverage, employing more sensors and using multiple hops transmission are the natural choice. The clustering technique we have designed for single-hop sensor networks suffers the following deficiencies: (1) the cluster formation step is solely based on spatial attributes, which causes poor performance for multi-hop sensor network where closely located sensors are more likely to be missing together, although it performs excellently on single-hop sensor networks where closely located sensor are not likely to be missing together; (2) The multi-hop sensor networks are usually targeted for complex, large-scale and dynamically changing phenomena where the relationship among the sensors changes over time; (3) The cluster formation restricts the search space for association rules; however in a dynamically changing environment, the static cluster formation step may suffer from not to have the related sensors in the same cluster; hence the cluster formation step should be dynamic and events aware; (4) In a multi-hop network, a failure of an intermediate sensor can cause a loss of multiple sensors' data; therefore if the clusters are fixed based on spatial attributes, there is a chance that all the sensors of a cluster would be missing together, which will result in the unavailability of the antecedent sensors to estimate the consequent sensor.

Motivated by the issues related to basic clustering techniques for multi-hop sensor networks, we describe an extension of our basic framework for multi-hop sensor network called MASTER-M. MASTER-M makes use of a dynamic clustering method that tackles the problems of simultaneously missing spatially correlated sensors and static location based cluster formation of spatially correlated sensors. The new clustering approach dynamically adjusts the clusters with the change of the relationships between the sensors. Moreover MASTER-M is more robust with respect to the number of simultaneously missing sensors.

MASTER-M groups the sensors into some clusters based on our proposed novel distance function described in the next subsection (3.3.1) to compute the distance between the sensors. The distance measurement in MASTER-M is derived in a bootstrapping fashion, i.e., the initial distance value is computed using the first few rounds of data, and the consequent distance value is updated incrementally. A re-clustering procedure is invoked once the distance bound in a

cluster does not hold any more. Another trigger for the re-clustering procedure is the user-defined number of rounds when a phenomenon change occurs.

1.2.1. The Clustering Metric

At the beginning, we arrange all the sensors according to their data missing rates in a descending order. The missing rate is defined as $\text{missing rate} = \frac{\text{number of missing rounds}}{\text{total number of rounds}}$. Let $(S_1, S_2, S_3, \ldots S_n)$ be the sorted list of sensors after sorting them in descending order of their missing rates, i.e., Sensor S_n misses the least often and sensor S_1 misses the most often. Sensors with the highest missing rates will be the "seeds" of the clusters. The significance of a seed is twofold. For a clustering technique, careful seeding is usually important and helpful [Arthur, 2007]. For data estimation, seeds are the most demanding nodes as they are most likely to miss. For each pair of sensors, S_i and S_j, we compute the distance between them. There are two types of distance between these two nodes: the standard deviation of the differences of the data readings, $d_{SDOD}(S_i, S_j)$, and the simultaneously missing rate, $d_{SMR}(S_i, S_j)$. $d_{SDOD}(S_i, S_j)$ shows the degree that S_i and S_j are related to each other. A relatively small $d_{SDOD}(S_i, S_j)$ implies a better correlation between S_i and S_j. $d_{SMR}(S_i, S_j)$ shows whether S_i and S_j tend to be missing simultaneously; a small $d_{SMR}(S_i, S_j)$ implies a small chance that S_i and S_j are missing together. So $d_{SDOD}(S_i, S_j)$ and $d_{SMR}(S_i, S_j)$ both are very important for deriving association rules between S_i and S_j and estimating missing sensor data. Note that both distances between a sensor and itself is always zero, i.e., $d_{SDOD}(S_i, S_i) = 0$ and $d_{SMR}(S_i, S_i) = 0$.

We further normalize $d_{SDOD}(S_i, S_j)$ and $d_{SMR}(S_i, S_j)$ to be the values between 0 and 1 and we name them $n_{SDOD}(S_i, S_j)$ and $n_{SMR}(S_i, S_j)$, respectively. These two distances form a two dimensional geometric space for a sensor node S_i where $n_{SDOD}(S_i, S_j)$ is placed along the x-axis and $n_{SMR}(S_i, S_j)$ is placed along the y-axis. Each data point in the two dimensional space formed for S_i represents a sensor node (S_j) where the abscissa is $n_{SDOD}(S_i, S_j)$ and the ordinate is $n_{SMR}(S_i, S_j)$. The origin is composed of the sensor itself, i.e., the point $(0, 0)$ represents the sensor (S_i). The Euclidean distance ($d(S_i, S_j) = \sqrt{n_{SDOD}(S_i, S_j)^2 + n_{SMR}(S_i, S_j)^2}$) is measured from the origin to S_j. The distance is then characterized as a measurement of the priority/benefit of putting S_i and S_j into the same cluster. Now we establish a matrix of distances from each node to all other nodes. Note that the distance relationship is symmetric, i.e., $d(S_i, S_j)$ and $d(S_j, S_i)$ are the same ($d(S_i, S_j) = d(S_j, S_i)$). Due to the symmetry of the distance function, we do not need the full matrix. The half matrix is defined as M,

$$M = \begin{pmatrix} 0 & d(S_1,S_2) & d(S_1,S_3) & \ldots & d(S_1,S_n) \\ & 0 & d(S_2,S_3) & \ldots & d(S_2,S_n) \\ & & 0 & \ldots & d(S_3,S_n) \\ & & & \ddots & \vdots \\ & & & & 0 \end{pmatrix}$$

```
procedure initialClusterSetup
1       construct a sorted list of the sensors according to their missing
        rates: DS = {S1, S2, S3, ..., Sn};
2       form a set of clusters C1, C2, C3, ..., Cn where Ci = {Si} for i=1
        to n;
3       loop until no change takes place
4         find the two closest sensors (Si, Sj) (without losing any
        generality we can assume i < j);
5         find the cluster Ci  where sensor Si belongs to;
6         find the cluster Cj where sensor Sj belongs to;
 7        if |Ci| + |Cj| < resource constraint (c)
8           merge (Ci, Cj);
9         end if;
10      end loop;
end procedure
```

Figure 3. The Initial Clustering Algorithm

1.2.2. The Initial Cluster Structure and Clustering Algorithm

Figure 3 shows the detailed algorithms for the initial cluster setup. The initial clustering algorithm starts with sorting the sensors according to their missing rates (line 1). In the next step we setup a set of clusters where each cluster contains only one sensor (line 2). In the third step the two nearest sensors that do not belong to the same cluster is identified (line 4) and their respective clusters are also obtained (lines 5 and 6). Merge the two clusters unless the sum of their size is greater than the resource constraint (*c*) [2] (lines 7 and 8). Step 3 is repeated until no merge operation can take place. Finally the algorithm outputs a set of clusters where each cluster contains no more than *c* number of sensors and two sensors in the same cluster are less likely to be missing together and more likely to be correlated.

1.2.3. Online Cluster Adjustment

In Figure 4 we describe the online cluster adjustment procedure. As each round of sensor readings (or each round for short) comes, we compute the distances between the reported values of each pair of sensors and compute the number of simultaneously missing sensors if there is any sensor missing. We compute $n_{SDOD}(S_i, S_j)$, $n_{SMR}(S_i, S_j)$ and $d(S_i, S_j)$ (lines 2, 3 and 4) for each pair of sensors S_i and S_j from the rounds arrived since the cluster has formed. In the next step, for each cluster we evaluate the distance between every two sensors inside a cluster. If the distance between any pair is greater than 1, we identify the current cluster as an obsolete cluster where the standard deviation of differences and/or simultaneous missing rate changed substantially; hence we need re-clustering. The value 1 signifies either the correlation or the simultaneously missing rate among sensors in a cluster reaches the maximum limit. Concurrently we check if the number of rounds reaches a user-defined ceiling as the user who has domain knowledge may anticipate phenomenon changes occurring and the need of re-clustering. The re-clustering is done by invoking the initial cluster setup algorithm (line 12). By online adjustment we maintain the most correlated sensors in a separate cluster and the sensors that are more likely to be missing together in other clusters.

```
procedure onlineClusterAdjustment (each data round)
1       for each pair of sensors  Si and Sj
2         compute nSDOD(Si, Sj)
3         compute nSMR(Si, Sj)
4         compute d(Si, Sj)
5       end loop
6       for each cluster
7         if the distance between any two sensors d(Si, Sj) is greater than 1
        or the number of rounds reaches the user defined number of rounds
        at which a phenomenon change occurs
8           needReCluster = true;
9         end if;
10      end loop;
11      if needReCluster
12        invoke initialClusterSetup();
13      end if;
end procedure
```

Figure 4. The Online Cluster Adjustment Algorithm

1.3. Implemented MASTER-M using C++ and conducted experiments comparing MASTER-M with two existing estimation algorithms for data streams, SPIRIT and TinyDB, using sensor data gathered from the Intel Berkley Lab sensor network and synthetic dataset.

Before explaining the results we briefly introduce the dataset we used to evaluate the performance of MASTER-M.

1.3.1. Intel Berkley Lab Data

This real-life application dataset is from the Intel Berkeley Lab. It contains environmental readings collected between February and April in 2004 in an indoor setting [Intel, 2009]. The dataset was collected using a multi-hop sensor network consisting of 54 sensors (Mica2Dot). Each sensor detects the temperature of the floor. The number of hops and the network topology for the dataset change dynamically as given by TinyDB [Madden, 2005]. The total number of rounds collected for all the sensors are approximately 65,000. Some random sensors' readings are missing in every round. Although the original dataset contains missing data, we cannot use the inherent missing data to evaluate the performance of the algorithms. This is because we do not know the correct values of the missing sensor readings; hence it is impossible to determine the accuracy of the algorithms. Therefore we cleaned the data in the first step and implanted the missing values into the cleaned dataset. Our cleaning process is iterative. Each round consists of sensor readings from all the sensors. If any of the sensors' readings is missing in a round, we removed the entire round. This is necessary because we process the data round by round. But we found that very few rounds can be obtained if we cleaned round by round; therefore in the second step we cleaned sensor by sensor. If a sensor is missing in more than fifty percent of the rounds we removed that sensor. Removing such a sensor will stop us removing the rounds where

only that sensor was missing. By repeating the entire process we ended up with nine sensors (sensor ids 41 to 49). We obtained three thousands rounds of data for those nine sensors. Since the network is a multi-hop sensor network, the dataset was used to evaluate MASTER-M which was designed for multi-hop sensor networks.

1.3.2. Factory Floor Temperature Data (Synthetic Dataset)

Besides the above real-life application dataset, we also synthesized a factory floor temperature dataset [Silberstein, 2006] which exhibits dynamically changing phenomena. We use this dataset to simulate a multi-hop sensor network, mobile sensor network and multiple server sensor network. In this simulation, machines are placed on a grid floor. In the beginning all machines are off and the initial temperature for all grid points is set to zero. The boundary grid point temperature is held at zero throughout the experiment. Some machines will be turned on for a number of rounds; the temperatures on those machines will reach a high constant temperature and heat will disperse on the floor. For each time step, at any non-boundary grid point(i, j), the temperature $T(i, j)$ is updated using the following formula (3):
$T(i,j) \leftarrow T(i,j) + alpha * [T(i+1,j) + T(i-1,j) - 2 * T(i,j) + beta * [T(i,j+1) + T(i,j-1) - 2 * T(i,j)]$
where alpha and beta are ≤ 0.25 and are the dispersion factors in the x and y directions, respectively. In this simulation, we simulated the scenario in which we sampled the sensor readings once per hour.

In total we gathered 4500 rounds of readings from 24 sensors for a multi-hop sensor network. For this dataset, the machines' on and off status reflects the thermal phenomena changes. Machines were placed at different locations and they were turned on randomly. As a set of machines were turned on, the heat transfer started from the turned-on machines to the boundary and the transfer process took place in a different direction. So the relationship among the different locations changed overtime; hence this dataset reflects the phenomena change, a property of many applications in multi-hop sensor networks.

1.3.3. Results for the Intel Berkeley Lab Dataset

The results (Figure 5) show the performance of our multi hop sensor protocol in terms of relative error (MAE) in the estimated value of the missing data. When the number of rounds of sensor readings is large, i.e. the amount of data used in the estimation process is large, MASTER-M performs much better than the other two algorithms although it is not the best one when the number of rounds is small. MASTER-M shows a very stable performance over time, while the other two methods perform very well at the beginning but deteriorates over time. The data distribution changes and different sensor readings vary differently over time; hence the estimation accuracy for TinyDB and SPIRIT drops. The stable performance of MASTER-M over time implies that MASTER-M is not vulnerable to concept drift – a phenomenon that occurs when the data distribution changes. As an approach applied on data streams, the long term trend is more important than the results obtained in the beginning stage, and MASTER-M shows its advantages.

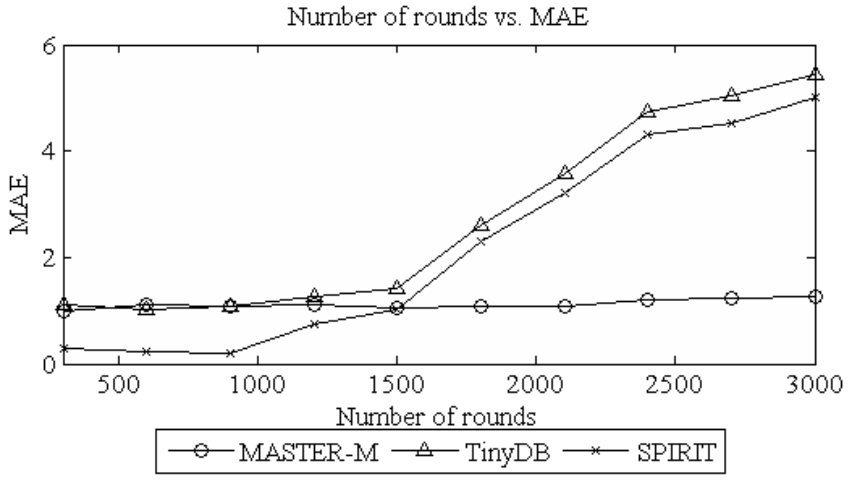

Figure 5. MAE vs. number of rounds

Table 1. Relative average error compared to MASTER-M

Approach	Average MAE
MASTER-M	1.11
TinyDB	2.70
SPIRIT	2.20

Table 1 shows the average MAE for all the three approaches and the relative average error for TinyDB and SPIRIT compared to MASTER-M. According to Table 1 MASTER-M has 58.89% less error than TinyDB and 49.55% less error than SPIRIT.

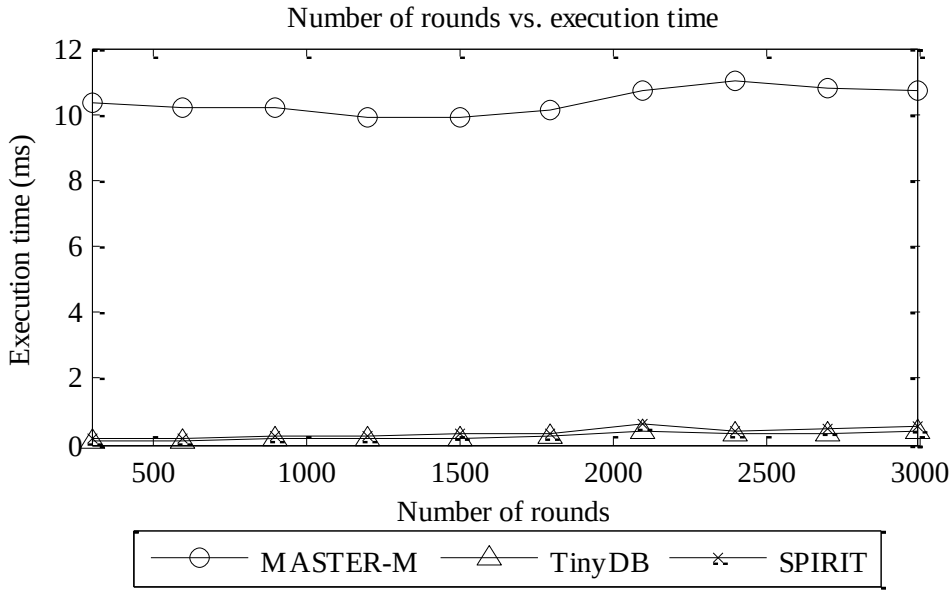

Figure 6. Number of rounds vs. execution time

In terms of execution time, our framework performs similarly on multi-hop sensor networks too. Like in single-hop sensor networks which we have reported in the previous years' annual reports, our framework takes more time on multi-hop sensor networks compared to TinyDB and SPIRIT,

but it offers very good estimation accuracy. Figure 6 shows the execution time of our approach with respect to the number of rounds. The execution time does not show much variation for any of the competitive approaches with respect to execution time. Hence, all the techniques perform similarly with respect to the number of rounds. The next section presents the results for the synthetic dataset of factory floor temperature.

1.3.4. Results for the Factory Floor Temperature Dataset

Figure 7 shows the MAE with respect to the number of rounds using the synthetic dataset. MASTER-M shows a constant performance over time even though the dataset includes phenomena changes. During the 4,500 rounds time period, the phenomena change many times, and each time MASTER-M correctly puts the related set of sensors into the same cluster; therefore, MASTER-M produces more meaningful association rules and hence better estimation accuracy. TinyDB and SPIRIT show a poor performance because they are not capable of estimating missing sensor readings when the sensor readings change randomly and there exist different relationships among the sensors at different points of time.

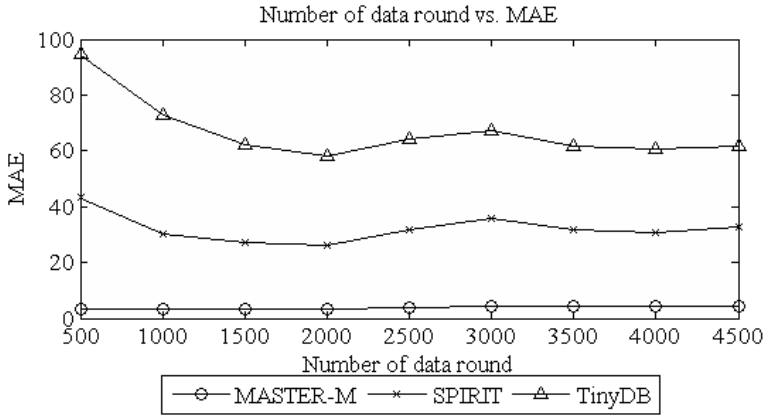

Figure 7. MAE vs. number of rounds

Table 2 shows the average MAE and average relative error for all three approaches. On average MASTER-M outperforms other two methods significantly.

Table 2. Relative average error compared to MASTER-M

Approach	*Average MAE*
MASTER-M	3.90
SPIRIT	32.2
TinyDB	67.1

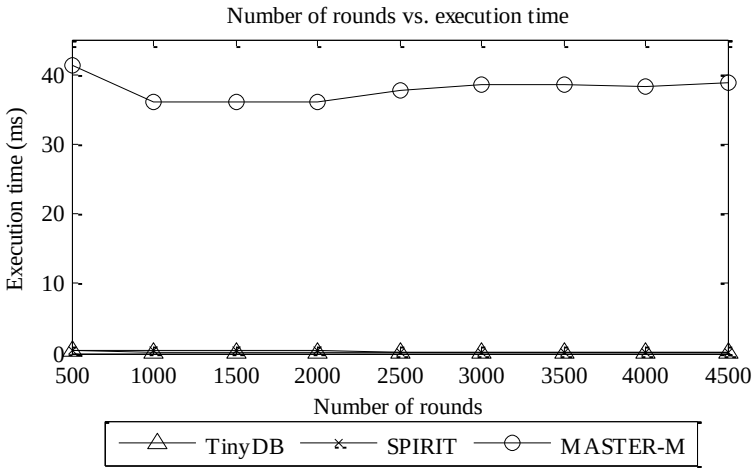

Figure 8. Number of rounds vs. execution time

The execution time for the sysnthetic dataset follows the same pattern as that for the real dataset. Figure 8 shows the execution time with respect to the number of rounds. Our approach takes longer execution time compared to TinyDB and SPIRIT, but offer very estimation accuracy. Moreover the execution time is less than 50 milliseconds which is significantly low compared to the arrival rate (typically in order of seconds). In the next section, we present the empirical results of our framework on mobile sensor networks.

1.4. Investigated additional satellite applications from NASA and gathered additional satellite data for further testing for MASTER-M.

1.4.1. DORIS Dataset

DORIS is a dual-frequency Doppler system that has been included as a host experiment on various space missions [DORIS]. The current missions with on-board DORIS receivers are TOPEX/Poseidon, Jason-1, Envisat, and SPOT-2, -4, and -5. Unlike many other navigation systems, DORIS is based on an uplink device. The receivers are on board the satellite while the transmitters are on the ground. This creates a centralized system in which the complete set of observations is downloaded by the satellite to the ground center, from where they are distributed after editing and processing. The system was developed to provide precise orbit determination and high accuracy location of ground beacons for point positioning. An accurate measurement is made of the Doppler shift on radiofrequency signals emitted by the ground beacons and received on the spacecraft. Some of the scientific uses of DORIS data include: (1) Precise orbit determination, (2) Maintenance of global accessibility to, and the improvement of, the International Terrestrial Reference Frame (ITRF), (3) Monitoring Earth rotation, etc. Daily DORIS tracking data since January 1992 (TOPEX), 1994 (SPOT-2,-3, -4, and -5), 2002 (Jason-1 and Envisat), 2008 (Jason-2) are available for free download. In this dataset missing data occurs due to numerous reasons including Orbit maintenance maneuver, Diode software failure, failure in high speed multiplexer, antenna damage due to hail, device malfunctioning, etc [DORIS-log]. For our experiment we use data from Envisat.

Each ground station transmits a number of attributes like Station ID, Measurement type, Time system indicator, Time observation, Meteorological data (Surface pressure, Surface temperature,

Relative humidity), Ionospheric refraction correction, Topospheric refraction correction, Meteorological data source, etc. For our experiments we use Meteorological data. We considered the entire system as a sensor network, where each ground station represents a sensor and the satellite works as a base station. A satellite is receiving data from each ground station periodically (once a day), and the sequence of data from each ground station forms a data stream. At any point in time, if the satellite fails to collect data from a ground station, it estimates the data using our technique. In this way missing ground station readings are filled by our technique.

1.4.2. Global Hydrology Dataset

Global temperatures have been monitored by satellite since 1979 with the Microwave Sounding Units (MSU) flying on the National Oceanic and Atmospheric Administration's (NOAA) TIROS-N series of polar-orbiting weather satellites. Data from nine separate satellites have been combined to provide a global record of temperature fluctuations in the lower troposphere (the lowest 5 miles of the atmosphere) and the lower stratosphere (covering an altitude range of about 9-12 miles) [MSU], [MSU-Desc].

The lower tropospheric data are often cited as evidence against global warming because they have as yet failed to show any warming trend when averaged over the entire Earth. The lower stratospheric data show a significant cooling trend, which is consistent with ozone depletion. In addition to the recent cooling, large temporary warming perturbations may be seen in the data due to two major volcanic eruptions: El Chichon in March 1982 and Mt. Pinatubo in June 1991. In this dataset a satellite might fail to provide data for unknown reasons [MSU-log] which is not explicitly explained publicly but [MSU-log] provides some situations when data is missing from one or more satellites. Hence the missing satellites are estimated by our approach.

An hourly precipitation data of sixteen locations from 1978 is freely available in [MSU-Data]. In our experiment we consider each location as one sensor (each location's information is truly coming from a satellite) and data collection center as base station. If a satellite fails to provide data, we assume some locations will be missing and the missing locations' readings are estimated by our approach.

1.4.3. Surface Meteorology and Solar Energy Dataset

This dataset is collected from Atmospheric Science Data Center that compiled it for Prediction of Worldwide Energy Resource Project [Energy-data]. The dataset contains data for 1 degree longitude by 1 degree latitude equal-angle grid covering the entire globe (64,800 regions). The NASA Goddard Earth Observing System (Version 4) generated the data using Multiyear Assimilation of Time series Data. The dataset has a spacing of 1.25 degrees of longitude by 1 degree of latitude. Bilinear interpolation is used to produce 1 by 1 degree regions. This dataset is assimilated by NASA through its Science Mission Directorate. The data is global and continuous in time [Energy-data]. The dataset was intended to provide easy access to the parameters needed for renewable energy industry.

The entire dataset measures many meteorological attributes including average air temperature, maximum air temperature, minimum air temperature, specific humidity, relative humidity, surface air pressure, etc. All these attributes are measured for 10 meter height. The dataset bears

the uncertainty for temperature, surface pressure, relative humidity and wind speed. The satellite derived values are considered to be more accurate than surface measured values. The Root Mean Square Error (RMSE) value for average temperature is 2.13, relative humidity is 9.40, and so on. Interested readers are referred to [Energy-data] for the entire list.

In our experiments, we considered each grid point as one data source / sensor; therefore at most we may have 180 x 360 data sources. The size of the dataset is huge and it is very difficult to work with the entire dataset simultaneously. To limit the size of the dataset, we performed our experiments on a single dimension namely Average Air Temperature. The daily average air temperature from 1983 to 2005 was collected for 100 grid points. The dataset contains some missing readings and we cleaned the missing readings and injected our simulated missing readings so that we can estimate the accuracy of our technique.

1.4.4. Results for the DORIS Dataset

The results (Figure 9) show that SPIRIT and MASTER-M are much better than the Average and TinyDB methods and they show a very stable performance over time, and SPIRIT is slightly better than MASTER-M. We find that first the sensor's change trend seldom repeats, and second, one sensor reading's change is almost independent from the other sensor readings' changes. Under these two conditions, MASTER-M cannot perform very well as it does not discover many association rules. As SPIRIT catches data readings change more quickly from the history data point only, for this dataset due to its ability of maintaining hidden variables on history data, it performs a little bit better than MASTER-M for this specific dataset.

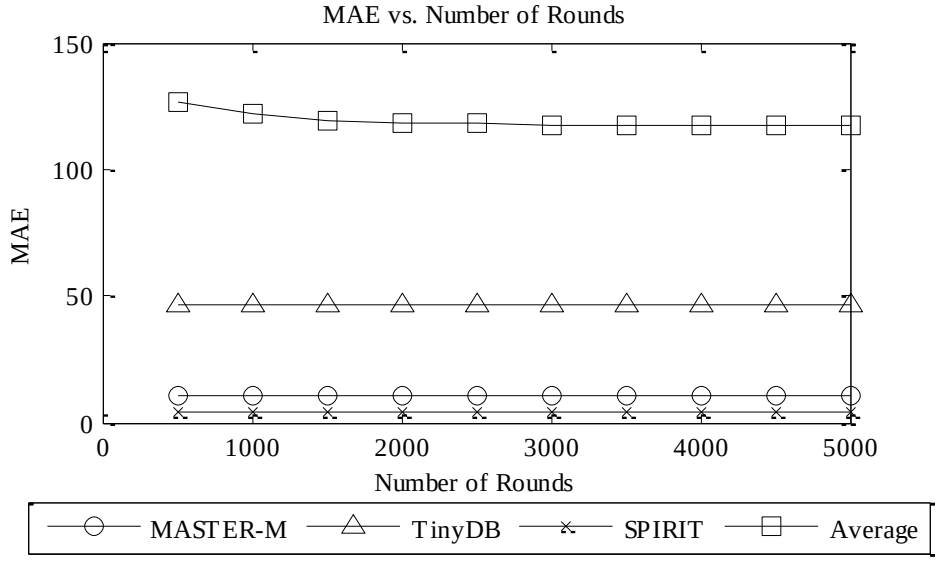

Figure 9. Number of rounds vs. MAE for DORIS data

1.4.5. Results for the Global Hydrology Dataset

Figure 10 shows the MAE with respect to the number of rounds using the Global Hydrology dataset. MASTER-M shows a stable performance over time even though the distribution of this dataset changes. It persistently outperforms on all other approaches in our experiments. In most

of the study period, the estimation error of the Average method keeps increasing, the estimation error of TinyDB is unstable and is the always the worst, the estimation error of SPIRIT is between those of Average and TinyDB and is much worse than that of MASTER-M. This experiment shows MASTER-M's advantages over other techniques. Average, TinyDB and SPIRIT show a poor performance because they are not capable of estimating missing sensor readings when the sensor readings change randomly and there exist different relationships among the sensors at different points of time. From the time series of precipitation, we came to know that the data change trends repeat and one sensor's reading change is similar to the other sensors' reading changes, i.e., the association rules among one sensor's readings with others are very strong. This environment is very typical for sensor networks. Under this environment MASTER-M shows the best performance.

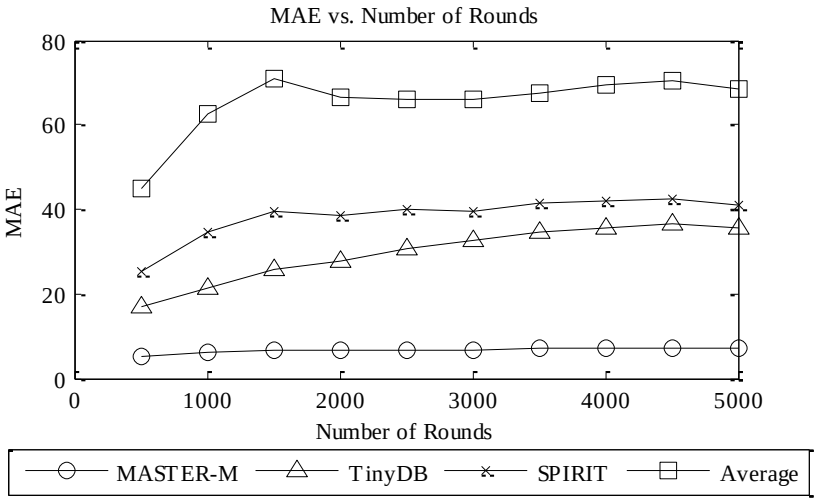

Figure 10. Number of rounds vs. MAE for Global Hydrology Data

1.4.6. Results of the Surface Meteorology and Solar Energy Dataset

Figure 11 shows the MAE with respect to the number of rounds using the Surface Meteorology and Energy dataset. In this experiment, SPIRIT performs the best and MASTER-M performs the second best among all the techniques (however, this was due to the bias toward auto-regression methods inherent in the dataset, which we will explain in the next paragraph). Their performance is very stable and much better than that of TinyDB, which is a little bit better than that of the Average method. To explain the results in a meaningful way, we analyzed the sensor readings.

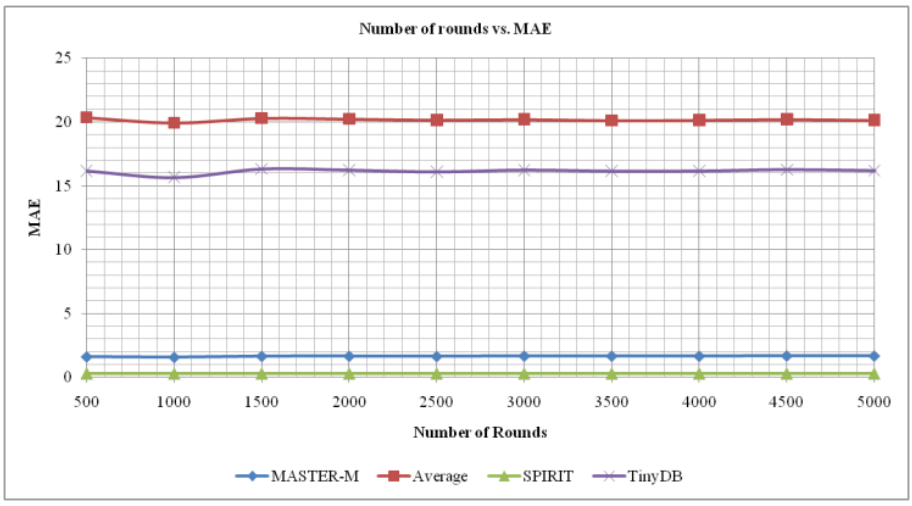

Figure 11. MAE vs. Number of rounds for Surface Meteorology and Solar Energy Dataset

From the sensors' readings' time series, we learned that these sensors readings display a kind of "linearly organized" property. In fact, it is true for this dataset [Energy-data], the raw dataset contains plenty of missing values and they are interpolated by some auto-regression method. In other words, a substantial part of this dataset is not from real sensors' readings, but from mathematically filled-up values. When we run experiments on this dataset, it favors the auto-regression method, which is SPIRIT in this experiment. To understand this, we can imagine using the same method filling up the missing values, then running itself again on the same data set with some generated missing values. In this way the result is biased and the auto-regression method performs much better than it should be. One important observation is that even with such a bias which requires application of interpolation twice to fill up missing values (one from a mathematically filled up method and one from SPIRIT), SPIRIT gives very small gain in accuracy compared to MASTER-M.

1.5. Completed the development of DEMS (A Data Mining Based Technique to Handle Missing Data in Mobile Sensor Network Applications), an algorithm to estimate missing sensor data and discovering knowledge in mobile sensor networks.

Our basic framework with necessary clustering for single-hop and multi-hop sensor networks that we have described in the previous sections works perfectly but fails for mobile sensor networks. So far in the clustering techniques (Sections 3.2 and 3.3), the cluster formation step is based on the spatial attributes or readings of a sensor. However, in a mobile sensor network, the spatial data of a sensor are changing and the relationships among the readings also change over time. Thus, the prior knowledge about sensor locations and data are not enough for mobile sensor networks. One possible solution for this problem is re-clustering whenever a sensor changes its location, but re-clustering is very computation-intensive and may cause loss of the history data, and thus loss of history data synopsis (the moments) stored in the MASTER-tree. Hence neither location-based nor data-based clustering for mobile sensors produces any meaningful result. Moreover, in a mobile sensor network, a reading of a sensor is accompanied by the sensor's location. So, if a sensor is missing, it is very likely that the reading and the location of that sensor will be missing together. Hence the estimation technique must estimate

both the dimensions for the missing sensors, which means that location prediction has to be an inherent part of the technique.

In our basic framework, association rule mining can be used to discover the relations among sensors. According to Tobler's first law of geography [Tobler, 1970], geographically close sensors are more correlated than the distant ones. In a mobile sensor network, the distance between the mobile sensors changes over time; therefore the correlation changes over time too. The association rules among the sensors represent the correlations among them. If the mobile sensors change their locations, the correlations among them change; hence the association rules previously obtained based on the sensor data will no longer be valid for the new locations. This has two-fold implications: (1) any previously explored rules may not be valid anymore; and (2) previously formed clusters may not be valid at all. In the extreme case, the history data from the same sensor may no longer be valid to estimate the missing data of the same sensor in the current round of data. This is because the old data are based on the previous locations of the sensor, whereas the new data are based on the new location. So the methods (e.g., Kalman Filter [Vijayakumar, 2009]) which use history data to estimate new data will also become invalid in such a situation.

Motivated by the related issues of the basic version of MASTER, we developed a new technique, called DEMS, for mobile sensor network applications. DEMS makes use of virtual static sensors that tackles the problems of location-aware clustering of real mobile sensors. It also tackles the problem of having no related history information for the current round of data from real mobile sensors. Moreover, DEMS addresses the issue of missing location of a real mobile sensor and is capable of predicting the next location for a missing real mobile sensor. The details of DEMS are presented in the next subsection.

1.5.1. The DEMS Approach

In DEMS, we exploit the spatial and temporal relations between sensor readings to estimate the missing sensor data. First we divide the entire monitoring area into hexagons based on a user-defined radius. Each hexagon corresponds to a virtual static sensor (VSS) placed at the center of the hexagon and covering the entire hexagon. A VSS is an artificial sensor, i.e., it does not exist physically in real life applications, but it exists in our technique as a synthetic sensor which mirrors a real static sensor. Each VSS has a unique identifier. DEMS converts the real mobile sensor readings into VSS readings based on the mobile sensors' current locations. Figure 12 shows *A* as the monitoring area covered by a MSN that is divided into 14 hexagons with 14 VSSs, $V_1 \ldots V_{14,}$ and 7 real mobile sensors, $M_{1..}$. $M_{7.}$

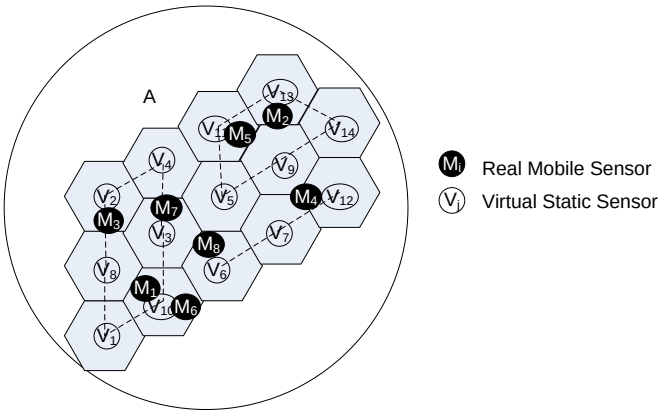

Figure 12. Monitoring area and hexagons

Using agglomerative clustering [Day, 1984], DEMS clusters the VSSs based on their locations into clusters and creates a MASTER-tree for each cluster. The dotted lines that connect the centers of the hexagons in Figure 12 show three clusters (V_1, V_2, V_3, V_8, V_{10}), (V_6, V_7, V_{12}) and (V_5, V_9, V_{11}, V_{13}, V_{14}). MASTER-tree records the data for the VSSs. For each missing mobile sensor reading, its estimated value is computed using the three major steps: 1) mapping the missing real mobile sensor to its corresponding VSS; 2) estimating the missing VSS reading using the discovered spatial and temporal association rules among the history VSS readings, and 3) converting the estimated VSS reading into the corresponding real mobile sensor reading.

In a MSN, a sensor reading reported is accompanied by the sensor location where the reading was obtained. Whenever a mobile sensor reading is missing (we call this a missing mobile sensor for short), it is likely that both the location and the reading will be missing together. To find the appropriate location of a missing mobile sensor we always keep track of mobile sensors' locations. A mobile sensor's location is mapped to a hexagon and the consecutive locations of a mobile sensor are mapped to a sequence of hexagons. We refer to a sequence of hexagons as a mobile sensor's trajectory. We mine the mobile sensor trajectories and predict the missing location based on the history trajectories. Morzy [Morzy, 2007] proposed a pattern tree based approach for mining trajectories and predicting future locations, which we adopt for DEMS. DEMS maintains a single pattern tree of trajectories for all the mobile sensors. As small devices like sensors often use the same protocol for relocation [Liu, 2005][Sibley, 2002], it is reasonable to assume that they have similar patterns of movement; therefore DEMS maintains a single pattern tree of trajectories for all the mobile sensors and uses a single pattern tree instead of an individual pattern tree for each mobile sensor. This trajectory pattern tree is used to predict a missing mobile sensor's location. The predicted location is used to map a mobile sensor to a VSS. Since sensors repeat the mobility pattern for relocation, history based trajectory mining is more promising than random walk models.

1.5.2. The Virtual Static Sensor

Sensor monitors a fixed region and a sensor's reading reflects an event occurring within this region; but in mobile sensor networks, owing to their mobile nature, the region being monitored varies with time. However, as in static sensor networks, the sensor readings for mobile sensor networks still reflect events occurring within a particular region. Our concept of virtual static sensors is directly motivated by the above fact. Every VSS, like sensors in SSNs, 'monitors' a

fixed region called its coverage area. An event occurring within a VSS's coverage area is reflected in its readings. However, unlike sensors in SSNs, VSSs do not have real existence and do not 'report' data to a base station. On the contrary, they are 'created' in our technique virtually to ease the spatio-temporal data mining.

A VSS reports a reading if there exists at least one real mobile sensor in the coverage area. A VSS is *active* if it reports in the current round and is *inactive* otherwise. VSS readings are readings of the real mobile sensor(s) which are present in the VSS's coverage area. In situations when multiple real mobile sensors are in a VSS's coverage area, the VSS reports the average of all the real mobile sensors' readings. There are two reasons for considering the average reading: (1) multiple sensors monitoring the same small coverage area most likely will report similar readings; and (2) any event occurring in the common coverage area will be reflected in the readings of all the sensors monitoring that area. As a hexagon is the atomic coverage region in DEMS, the radius of each hexagon is usually small enough to assure the variance of real sensors' readings from the same hexagon to be minimal, and averaging all readings from sensors from the same hexagon will be close to the real value of the corresponding region. A VSS is called a *missing VSS* if one real mobile sensor exists or expected to exist within the coverage area of that particular VSS and the reading from the real mobile sensor is missing.

```
Procedure mapReal2Virtual(RealSensorData listRSData,
VirtualSensorData listVSData)
1    for each real sensor rs
2      if(rs is not missing)
3        location ← listRSData(rs).Location
4        vs ← findVirtualSensor(location)
5        listVSData(vs).addReading(listRSData(rs).Reading)
6      else
7        location ← predictLocation(rs)
8        vs ← findVirtualSensor(location)
9        listVSData(vs).status←missing
10   end loop
11   for each virtual static sensor vs
12     if(listVSData(vs) has data)
13       listVSData(vs).status←active
14   listVSData(vs).reading←average(listVSData(vs).Readings)
15     else
16       if(listVSData(vs).status is not missing)
17         listVSData(vs).status ←inactive
18   end loop
end procedure
```

Figure 13. Mapping mobile sensor readings to virtual static sensor readings

Hence VSS readings are directly stored in our MASTER-tree. So, in DEMS, the MASTER-tree represents the relationships among the VSSs. We assume that at any instance, all the mobile sensors report their readings to the base station, which is then, mapped to the corresponding VSSs. Figure 13 shows the mapping algorithm in details. For each real mobile sensor, DEMS

finds the appropriate VSS (lines 3 and 4) using a geometric mapping between location and hexagon. If the location of the real mobile sensor is missing, DEMS predicts the expected location for the real mobile sensor and maps it to the appropriate VSS for that predicted location. If the mobile sensor reading is missing, DEMS marks the corresponding VSS as missing. Finally, in the loop from lines 11 to 18, each VSS is marked appropriately as active, inactive or missing. At any particular time, only the active virtual static sensors are stored in their appropriate MASTER-trees.

1.5.3. The Data Estimation Module

The DEMS does estimation in two steps as opposed to our basic framework, (1) estimate the VSS value and (2) calculate the value for missing real mobile sensor from corresponding estimated VSS value. Initially, the location of the missing mobile sensor is predicted based on the user-defined minimum support and minimum confidence using Morzy's approach [Morzy, 2007]. If the algorithm fails to predict the next location, DEMS uses the last reported location of the missing mobile sensor as its current location. Location prediction is preceded by mapping the missing mobile sensor to the corresponding VSS, which is called missing VSS. The estimated missing mobile sensor reading is the estimated missing VSS reading computed from the MASTER-tree. The estimated VSS's reading is directly used as the estimated reading for the missing mobile sensor.

1.6. Implemented DEMS using C++ and conducted experiments comparing DEMS with three existing estimation algorithms for data streams, SPIRIT, Average and TinyDB, using sensor data gathered from the DAPPLE project and synthetic dataset

This section starts with the brief description for the dataset followed by the detailed results we obtained for each dataset.

1.6.1. The DAPPLE project dataset

The real life dataset is obtained from the DAPPLE project [Dapple, 2004]. The data are about carbon monoxide (CO) readings collected over a period of two weeks around Marylebone Road in London. The mobile sensors monitoring the atmospheric CO level are attached to PDAs which store these readings. The data sampling rate of the sensors is once every second. The software on the PDAs generates log files containing the atmospheric pollution levels with the locations and the timestamps associated with the readings. Each reading was carried out with a single sensor kit every second for duration of about 45 minutes over a two-week period. Simultaneous use of multiple sensors (usually three) was limited to some days only. For our experimental purposes, we considered the instances when three sensors were simultaneously recording CO pollution levels for a considerable period of time. We chose Thursday, 20th May 2004, when three sensors were simultaneously recording for about 32 minutes, resulting in 600 rounds (after disregarding the missing rounds) of CO readings. Since the sensor nodes are moving, this dataset was used to evaluate DEMS which was designed for mobile sensor networks.

1.6.2. Factory Floor Temperature Data (Synthetic Dataset)

Besides the above real life application datasets, we also synthesized a factory floor temperature dataset [Silberstein, 2006] which exhibits dynamically changing phenomena. We use this dataset

to simulate multi-hop sensor network, mobile sensor network and multiple server sensor network. In this simulation machines are placed on a grid floor. In the beginning all machines are off and the initial temperature for all grid points is set to zero. The boundary grid point temperature is held at zero throughout the experiment. Some machines will be turned on for a number of rounds; the temperatures on those machines will reach a high constant temperature and heat will disperse on the floor. For each time step, at any non-boundary grid point(i, j), the temperature $T(i, j)$ is updated using the following formula (3): $T(i,j) \leftarrow T(i,j) + alpha * [T(i+1,j) + T(i-1,j) - 2 * T(i,j) + beta * [T(i,j+1) + T(i,j-1) - 2 * T(i,j)]$

where alpha and beta are $\leq$ 0.25 and are the dispersion factors in the x and y directions, respectively. In this simulation, we simulated the scenario in which we sampled the sensor readings once per hour.

To induce mobility for simulating a mobile sensor environment, we created a 100 mobile sensors roaming around in random directions to monitor the factory floor and report the temperature readings from different locations at different points in time. In our simulations, we sampled the mobile sensor readings once per hour. In total we gathered 5000 rounds of readings from 100 sensors. The dataset without mobile sensors was used to simulate multi-hop sensor network and mobile sensor network as we have described in the previous Section 1.3.

1.6.3. Result for the DAPPLE Project Dataset

Figure 14 shows the change of MAE with the change of the number of rounds of sensor readings. The MAE value of 0 for DEMS implies that DEMS estimates the missing data with no error. A possible reason is that the DAPPLE project dataset has very few variations (the CO levels are within the range 0~6) and the sensors have very high spatial correlations. In most cases the readings in the same cell are the same. Hence, DEMS produces a zero error in terms of MAE. The MAEs for other approaches are comparatively high at the beginning and become stable at the end as the number of rounds increases.

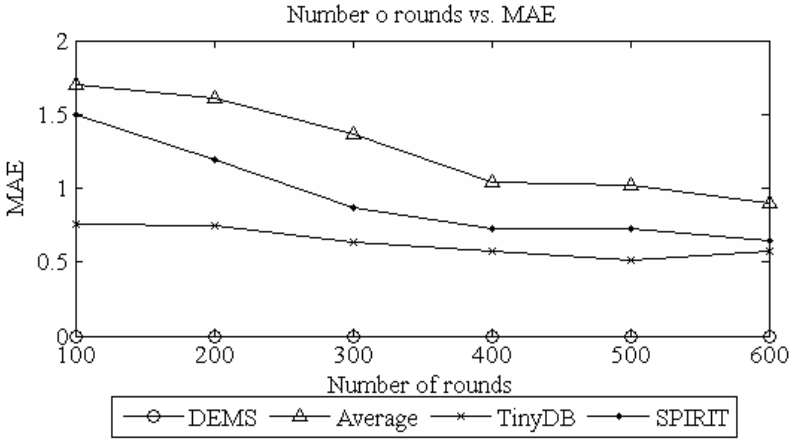

Figure 14. Number of rounds vs. MAE

Table 3 shows the average MAE for all the approaches. DEMS almost perfectly estimates the missing values while Average gives the highest error compared to SPIRIT and TinyDB.

Table 3. Average MAEs

Approach	Average MAE
DEMS	0
Average	1.2717
TinyDB	0.6331
SPIRIT	0.9437

Figure 15 shows the execution time of our approach compared with that of the other techniques. Presumably, our approach takes more time than the other three approaches but it offers almost perfect accuracy. However, the DEMS' execution time is only between 10-15 milliseconds and the data gathering frequency in the mentioned application is no less than 30 seconds. Hence the time is practically acceptable for many sensor network data stream applications.

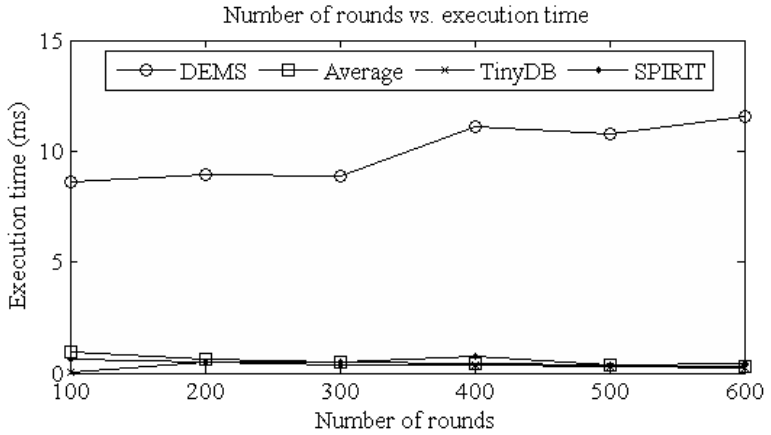

Figure 15. Number of rounds vs. execution time

1.6.4. Results for the Factory Floor Temperature Dataset

We performed a similar study for the factory floor temperature dataset. This dataset have more variations (temperatures are in the range 0~100C) compared to the DAPPLE project dataset. Figure 16 shows the change of MAE with respect to the change of the number of rounds. The MAE for each approach remains almost constant when the number of rounds changes. As this dataset has more variations than the DAPPLE project dataset, even though DEMS still performs better than the other techniques, its performance is not as good as that with the DAPPLE project dataset.

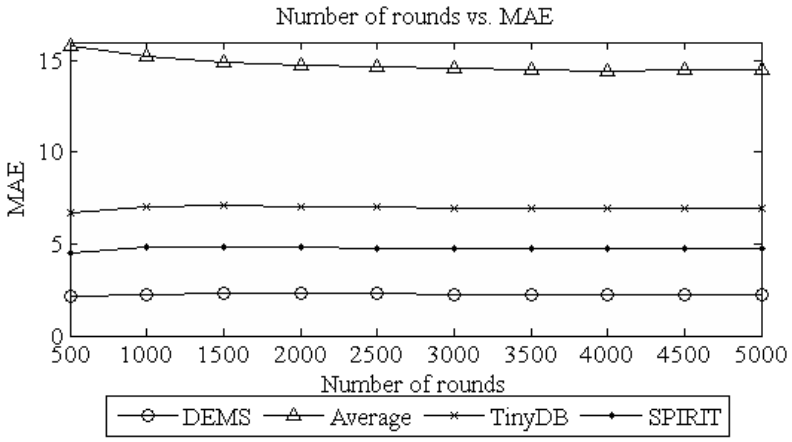

Figure 16. Number of rounds vs. MAE for the factory floor temperature dataset.

Table 4. Average MAEs for the factory floor temperature dataset

Approach	**Average MAE**
DEMS	2.2538
Average	14.7787
TinyDB	6.9621
SPIRIT	4.7472

Table 4 shows the average MAE for all the approaches. The average errors produced by Average, SPIRIT and TinyDB are about seven times, three times, and two times more than that produced by DEMS, respectively. DEMS is thus very effective in estimating missing sensor data. Figure 17 shows the comparison study of execution time with respect to the number of rounds for each approach. Like with the DAPPLE data set, here our approach also takes more execution time than the other three approaches and the execution time increases with the increase of the number of rounds.

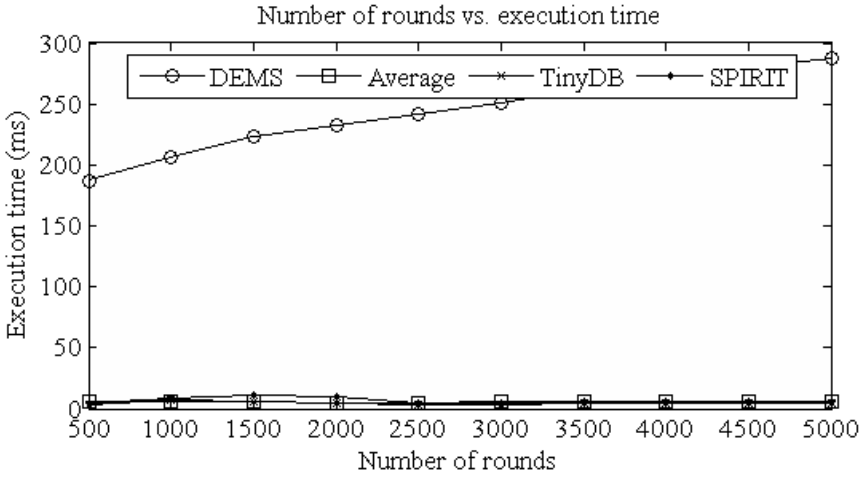

Figure 17. Number of rounds vs. execution time

1.7. Simulated multiple server sensor networks, extended our general framework for multiple server sensor network, and compared the performance of our framework with three existing techniques, SPIRIT, Average and TinyDB.

To simulate a multiple server sensor network, we used the factory floor temperature dataset and simulated the readings for 100 sensors and place 5 different servers on the floor. The load was distributed equally, which means almost 20 sensors report to one server. We ran the data estimation algorithm in each local server. Each sensor reports to the closest local server. We synthesized a total of 5000 rounds of data for each sensor. Missing data was injected randomly into each sensor to evaluate the efficacy of our technique.

1.7.1. Results for Multiple Server Sensor Networks

We run each data estimation algorithm in each server. From Figure 18 to Figure 22 we show our experimental results in terms of MAE vs. number of rounds for each of the servers 1 to 5, respectively, and Figure 23 shows the average MAE vs. number of rounds for all servers. For various local servers, MASTER-M consistently performs better than all the other techniques, showing it is a feasible data estimation technique for multiple local servers sensor networks applications. In fact, there is no substantial difference between the results in a single server and multiple local servers sensor networks applications. As in single server applications, MASTER usually performs the best, it is natural that MASTER outperforms all the comparing methods on this multiple local servers sensor networks application dataset.

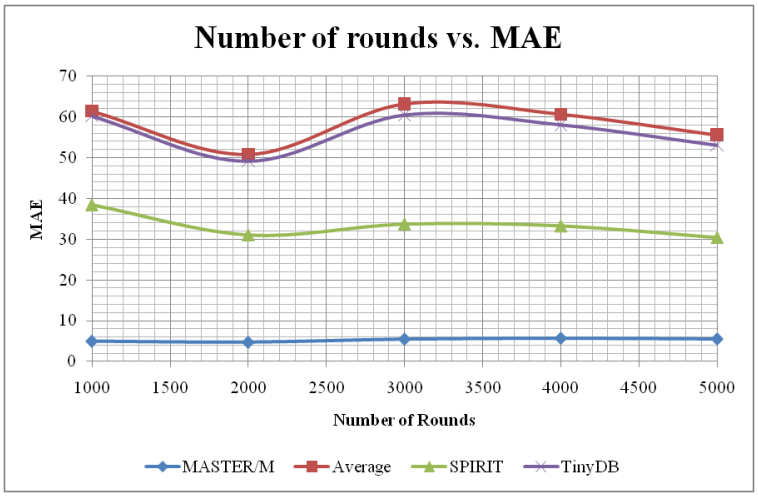

Figure 18: MAE vs. Number of rounds for Server 1

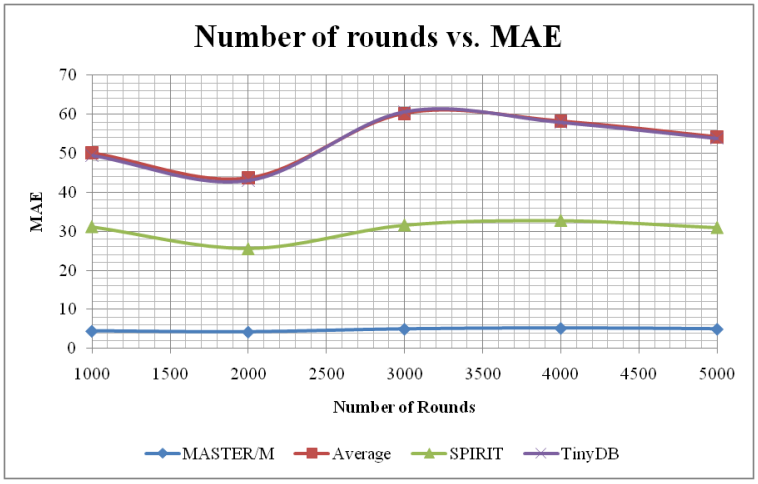

Figure 19: MAE vs. Number of rounds for Server 2

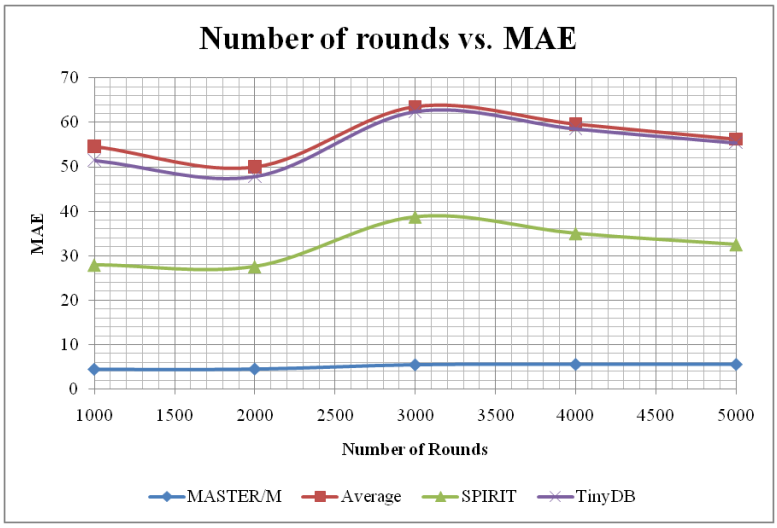

Figure 20: MAE vs. Number of rounds for Server 3

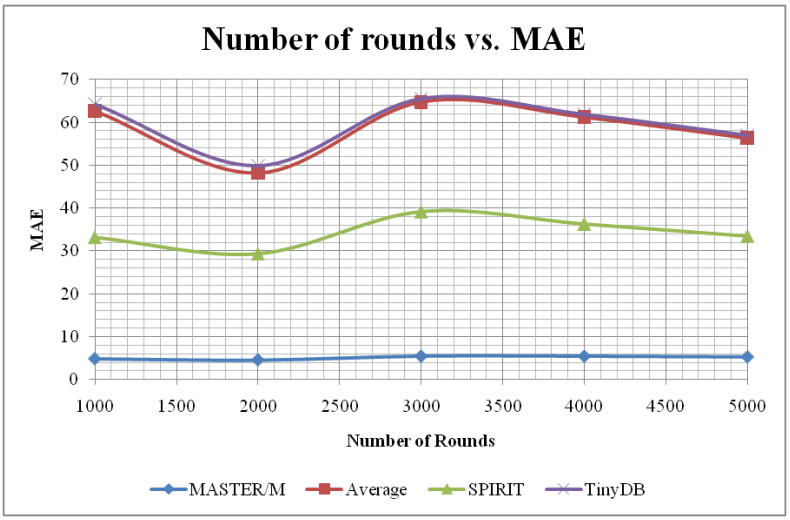

Figure 21: MAE vs. Number of rounds for Server 4

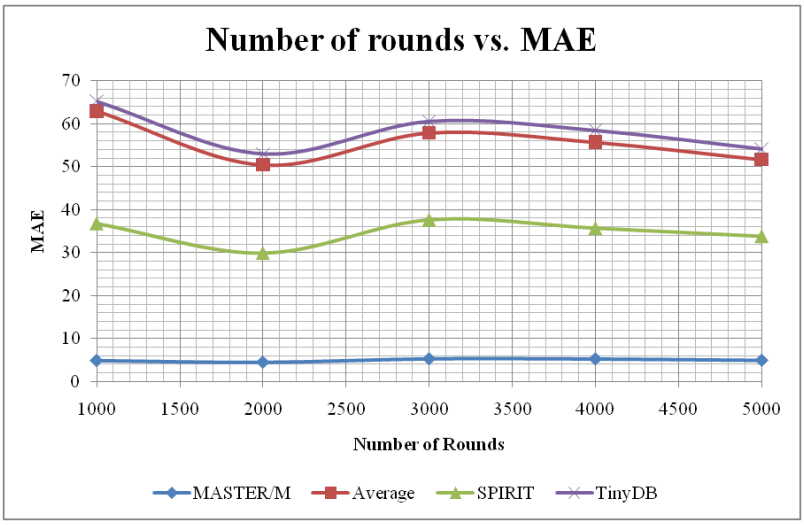

Figure 22: MAE vs. Number of rounds for Server 5

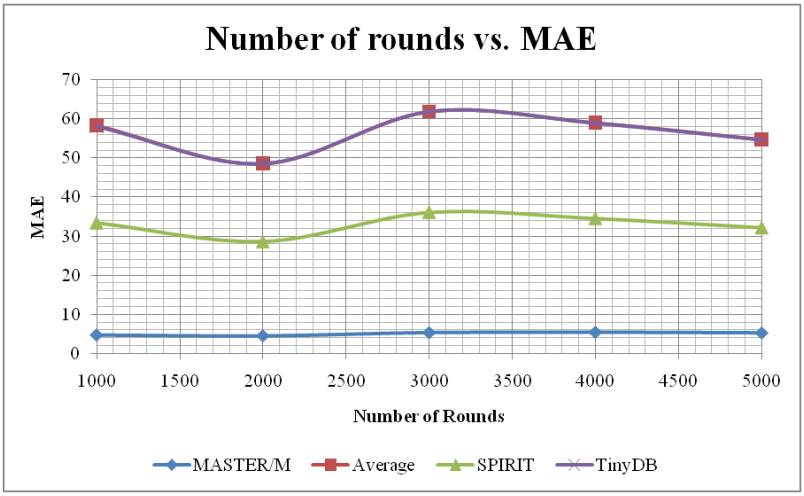

Figure 23: MAE vs. Number of rounds (averaging all servers)

1.8. Collected the spectral dataset from Dr. Nikunj C. Oza, our collaborator at NASA Ames Research Center, and performed experiments to evaluate our technique's performance on the spectral dataset.

Working with Dr. Nikunj C. Oza, our collaborator at NASA Ames Research Center, we were able to obtain the spectral dataset available at NASA and understood its meaning and application to apply it to our technique for performance evaluation. The spectral dataset consists of measures of reflected light for different wavelengths (typically 5 to 32) [Srivastava, 2004]. Sunlight reflected from different objects at different wavelengths (called channel) from earth is captured for remote sensing applications. Two types of instruments are used for spectral datasets called Advanced Very High Resolution Radiometer (AVHRR) and Moderate Resolution Imaging Spectroraiometer (MODIS). AVHRR can generate only 5 channels whereas MODIS can generate up to 32 channels. AVHRR data is available since 1981 but MODIS data is

available since 1999. The dataset we have received has 5 channels for AVHRR and 6 channels for MODIS. The reflections were calculated for 1740 x 1860 grid points from Greenland ice sheet and surrounding ocean. The difference between neighbor grid points is tentatively 1.25 kilo meter [Srivastava, 2004]. The daily measurements are collected for each grid point and compiled in a file. We have the measurements for middle of the year 2000 only. Only the middle of the year produces meaningful measurement because the data was collected from very near to North Pole which gets good sun exposure during summer and very poor sun exposure during winter per Dr. Oza.

In brief we have a very limited amount of data with respect to time domain, but we have a huge amount of data with respect to geographical location. Unlike other datasets, we consider each channel as one data source or sensor. Therefore, each grid point has 5 or 6 sensors. The reflected light at any point of time from one grid point is considered as one round. Therefore, one round consists of measurements for different wavelengths from the same location. Reflected lights at different points of time are considered as different data sources. In the rest of our datasets we consider each grid point as one source and a round is the data coming from all the sources at any point of time, but in this dataset we reverse the role of space and time. In this dataset we consider each channel at different point of time as one data source and measurements from the same location as a round of sensor readings.

In the dataset, each channel reports in a different range of values, and for experimental purposes, we normalize them all into 0 to 100 ranges (suggested by Dr. Oza). Each data file contains plenty of zeros from some location which represents the fact that there is no data available for the location [Srivastava, 2004]. For our experimental purposes, we remove a round if all the channels are zero. We inject artificial missing data into three datasets to evaluate our technique. We perform the experiments on the dataset which contains synthesized missing data.

1.8.1. Results for the Spectral Dataset

In this dataset, MASTER performs the best, while Average and TinyDB perform the worst. SPIRIT's performance is between the best and the worst. This dataset is special due to the following reasons: first, as we described before, for our experimental purposes, we switched the dimension of time and space; second, there are many really missing values which are marked by zeros in the dataset and we removed all of them before we run our experiment. For these two reasons, the relationships among the data in this dataset become very complicated which differs from regular time series data. However, as these data are still related to each other, MASTER still produce satisfying accurate estimation for missing values as it is able to discover arbitrary relationship among data items. SPIRIT, while performs well for many regular time series datasets, cannot obtain highly accurate estimation results on this dataset. From this experiment, we can see another example that shows MASTER's advantages when applied for stream data missing values estimation. Figure 24 shows the MAE resulting from the four techniques. MASTER gives the best MAE.

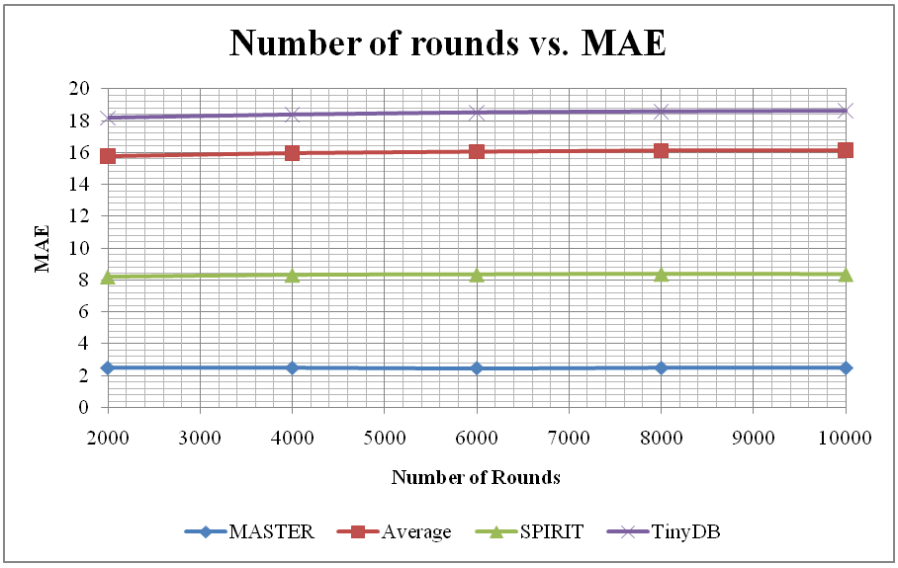

Figure 24. MAE vs. Number of rounds for the Spectral dataset

1.9. Provided a theoretical analysis of space and time complexity for MASTER trees

We divide our theoretical analysis into two parts: space and time. Since our basic framework consists of only MASTER-trees, here we only discuss the complexity analysis of MASTER-trees.

1.9.1. Space Complexity

In this section we analyze the space usage of our technique. The space taken up by any particular MASTER-tree is dependent on the number of the total cells that are allocated. Since cells are allocated adaptively, any vector space grid will not necessarily have all of its cells allocated. However, for the simplicity of analysis and as a first step, we will assume that each grid is entirely filled up. This will allow us to account for the worst case scenario and the maximum amount of memory that can potentially be allocated. Such formulation assumes the theoretical upper bound as an actual cost.

The space usage is the same as the total number of cells the MASTER-tree has and the number of cells depends upon the grid discretization scheme. For a chosen discretization, let the total number of cells allocated by any grid be G. To compute the total space consumption of one MASTER-tree, it is sufficient to calculate the number of grids needed. Each grid by assumption is full; each cell of a grid is a parent for a grid at the next tree level, and so on down to the leaf node; it follows that for a generic node at level l, we will have G^lgrids. The total number of grids that constitute one MASTER-tree for one cluster is therefore given by:

$$Space_{Tree}(c) = \sum_{l=1}^{c} n_l(c) G^l$$

where $n_l(c)$ is the number of node at level l and c is the number of sensors in the cluster. The maximum level a MASTER-tree can have is equal to the number of sensor nodes in a cluster.

Since MASTER-tree captures all possible relations among the sensor nodes, any sensor node appears as a descendant of all unique proposer subset of the sensor nodes. Therefore, for the cluster with c sensor nodes, at level c, any node can have $c-1$ predecessors, therefore can have $\binom{c-1}{c-1}$ unique paths from the root. Similarly at level $c-1$, any node can have $C-2$ predecessors, therefore can have $\binom{c-1}{c-2}$ unique paths from the root to it. Thus the total number of nodes at level l is $n_l(c) = c\binom{c-1}{l-1}$ which follows

$$Space_{Tree}(c) = \sum_{l=1}^{c} c\binom{c-1}{l-1} G^l$$

Now if we have total n number of sensors and c number of sensors in each cluster, the total number of clusters would be $\lceil n/c \rceil$. Moreover, each cluster may have more than one temporal snapshot, if each cluster has K number of temporal snapshots, the total space usage becomes

$$Total\ Space = K\lceil n/c \rceil \sum_{l=1}^{c} c\binom{c-1}{l-1} G^l$$

If d is the number of dimensions and m is the number of quantizations for each dimension, then $G = md$. Thus the entire space usage becomes

$$Total\ Space = K\lceil n/c \rceil \sum_{l=1}^{c} c\binom{c-1}{l-1} (md)^l$$

1.9.2. Time Complexity

In this section we analyze the time complexity of the update algorithm for our MASTER-tree data structure. Update time is the time required to update the set of MASTER-trees with one round of data. As each round arrives, we update the MASTER-trees with the new round of data. At any particular round, we update one MASTER-tree for each cluster. If n is the total number of sensors and c is the number of sensors in one cluster, then $\lceil n/c \rceil$ is the total number of MASTER-trees we update in each round. The update time therefore is

$$Update\ time = \lceil n/c \rceil T(c)$$

where $T(c)$ is the time required to update a MASTER-tree with c sensors. The time required to update the MASTER-tree is directly proportional to the total number of nodes we have in one MASTER-tree, $T(c) \infty N(c)$ where $N(c)$ is the total number of nodes in a MASTER-tree with c sensors. First we compute the number of nodes in a MASTER-tree without the grid structure. The total number of nodes is equal to the sum of the number of nodes at each level. Let $n_l(c)$ be the number of nodes at level l when the cluster has c number of sensors. Hence, the number of nodes $N(c) = \sum_{l=1}^{c} n_l(c)$ where $n_l(c)$ is the number of nodes at level l. According to the previous section (Section on Space Analysis) $n_l(c) = c\binom{c-1}{l-1}$, which follows $N(c) = \sum_{l=1}^{c} c\binom{c-1}{l-1}$. The update time therefore becomes

$$Update\ time = O\left(\lceil n/c \rceil \sum_{l=1}^{c} c\binom{c-1}{l-1}\right)$$

1.10. Estimated a theoretical energy savings for data estimation over retransmission

In this section, we study the energy savings when using data estimation in comparison to simple re-transmission of missing readings. We evaluate the energy consumption for single hop and multi hop sensor network environments. For single hop mobile sensor networks, as mobility induces extra energy consumption, the total energy consumption is akin to single hop static sensor network with an added mobility induced energy loss. Therefore the total energy consumption for a single hop mobile sensor network will be always greater than that for a single hop static sensor network. Therefore it suffices to say that any energy consumption in a mobile senor network will be a greater than in a single hop static sensor network by a factor α where α is the energy consumption due to sensor mobility. Hence energy evaluation for mobile sensor network is not included in our present work.

For a single hop sensor network, [Halatchev, 2005] proposed an energy evaluation technique based on comparing the total energy used by the sensors when using a data estimation protocol against simple re-transmission of the missing sensor data. In [Halatchev, 2005], energy consumption is based on the question that given a μ amount of sensor battery power initially, how many more rounds of data transmission is possible when sensors do not have to re-transmit their missing data. They calculated that on average, the extra number of rounds transmitted when using the data estimation protocol is about 2.5 times. The basic idea is that when a sensor network is using a data estimation technique for predicting missing sensor readings, it does not require the base stations to re-send the missing sensor readings. The base station can predict what the missing data is/was. However, without such a protocol, the sensors have to wait, listen and then resend the readings that failed to reach the base station. This entails extra work in terms of using energy resources at the sensors reducing their life cycles considerably [Halatchev, 2005]. As the proposed energy evaluation technique is independent of the data estimation algorithm running at the base station, but based on the network topology, given a similar network (static single hop sensor network), the results obtained for [Halatchev, 2005] is equally valid. As the network envisaged by [Halatchev, 2005] for their calculations is similar to our single hop sensor network, we claim that the results obtained by him will hold true for our single hop sensor networks also. Moreover, the aim of energy evaluation is to show the advantages in using data prediction protocols in terms of energy consumption, the experiments they performed validated it. Hence we did not perform energy evaluations for single hop sensor networks.

For multi hop networks, Heinzelman [Heinzelman, 2000] proposed a power calculation equation (PCE) where the amount of energy used in transmitting a sensor reading is directly proportional to the number of bits and the distance over which they are transmitted. It considers a network of *n* sensors arranged linearly and gives the power consumed by the network in transmitting *k*-bit data from the *nth* sensor to the base station. It also incorporates the energy used by the intermediate hops (sensors between the data originating sensor and the base station) in receiving and forwarding the data to the base station. In contrast, our modified energy calculation formula, given by Equation (1), calculates the energy consumed (*En*) using the actual distance (*ri*) between the sensors.

$$En = n\,(Etransmit \times k) + Eamplifier \times (r12 + \ldots + rn2) \times k + (n-1) \times (Ereceive \times k) \ldots\ldots\ldots \quad (1)$$

where *n* is the number of hops, including the sensor where the data originates, through which the data pass before reaching the base station; *k* is the number of bits transmitted; *ri* is the distance of the *ith* hop; *Etransmit* and *Ereceive* are the amount of energy consumed in running each transmit and receive circuitry, respectively; and *Eamplifier* is the energy dissipated in amplification circuitry for achieving acceptable transmission capability [Heinzelman, 2000]. The values used for each of the above mentioned parameters are given in Table 5 below which is the same as the ones assumed in [Heinzelman, 2000] .

Table 5. List of constants for the energy equation

Etransmit	***50 nJ/bit***
Ereceive	50 nJ/bit
Eamplifier	100 pJ/bit/m2
k	2000 bit

For a multi hop sensor network, we divided our network into sub-networks based on the actual physical sensor locations. The sub-networks consist of linearly arranged sensors so that each sub-network can be considered a linear network as given by [Heinzelman, 2000. The PCE is now readily applicable to each of the sub-networks. Using Equation (1), we calculate the energy consumed in transmitting a *k* bit data originating from each of the *n* sensors in an individual sub-network. Then, the total energy consumed by that individual sub-network with *n* sensors, in one data round of transmission, is given by Equation (2). Finally, the summation of the total energy consumption by each sub-network gives us the total transmission energy cost of a multi-hop sensor network.

$$Total\ Epower = \sum_{i=1}^{n} E_i \quad \ldots\ \ldots\ \ldots\ (2)$$

However, using a simple re-transmission of the missing sensor data instead of data estimation, all the sensors, after one transmission, are in the receiving mode for a possible re-transmission request from the base station. Hence in such a scenario, all the sensors are using energy to stay 'awake'. Here, we assume that the re-transmission requests involve a single re-transmission of the missing data. Then, the total energy (*EnM*) consumed in this case is given by Equation (3) where *t* is the duration for which a sensor must be in 'awake' mode for possible re-transmission requests.

$$EnM = n(Etransmit \times k) + Eamplifier \times (r12 + \ldots + rn2) \times k + (n-1) \times (Ereceive \times k) + t \times (Ereceive \times k) \ldots (3)$$

Thus, Equation (3) gives us the total energy consumed in transmitting *k* bit data originating from each of the *n* sensors in an individual sub-network using a simple re-transmission process. Next, the total energy consumption by each of the sub-networks and the entire network as a whole is calculated using Equation (2). The difference in total energy consumption in transmission when using the data estimation (Equation 1) and when using a simple re-transmission (Equation 3) gives us the amount of energy saved using data estimation. From our experiments, the energy savings amount to 20% which is significant considering that we fixed the missing data rate at 20% and limiting to single re-transmission of the missing data. Through experiments, we conclude that greater the percentage of missing data in a network, greater the energy consumed

by the network in re-transmissions, and greater the energy savings produced by data estimation. This justifies our stated argument for developing data estimation techniques rather than using simple re-transmissions.

In summary, our evaluation of energy consumption shows that using data estimation saves energy by avoiding re-transmission. There is a linear correlation between the percentage of missing data and the percentage of energy savings by data estimation techniques. Data estimation techniques save more energy with increasing percentages of missing data. In our energy calculation, we did not consider the subsequent missing sensor readings after a single re-transmission which will require even more energy than the one we show for a single re-transmission.

2. PUBLICATIONS TO DATE

- Nan Jiang and Le Gruenwald, "Research Issues in Association Rule Mining for Data Streams," SIGMOD RECORD, Vol. 35, No. 1, March 2006.
- Biao Liu, "Classify Data Streams using Concept-Drifting Detection Indicator," Master's Thesis, School of Computer Science, University of Oklahoma, May 2006.
- Nan Jiang and Le Gruenwald, "CFI-Stream: Mining Closed Frequent Itemsets in Data Streams," in the proceedings of ACM International Conference on Knowledge and Data Discovery (KDD), August 2006.
- Nan Jiang and Le Gruenwald, "An Efficient Algorithm to Mine Online Data Streams," in the proceedings of International Workshop on Temporal Data Mining, August 2006.
- Le Gruenwald, Hamed Chok, and Mazen Aboukhamis, "Using Data Mining to Estimate Missing Sensor Data," in the proceedings of IEEE Workshop on Optimization-Based Data Mining Techniques with Applications in conjunction with IEEE International Conference on Data Mining, October 2007.
- Frank Olken, Le Gruenwald, "Data Stream Management: Aggregation, Classification, Modeling and Operator Replacement," IEEE Internet Computing, Vol. 12, No. 6, November/December 2008.
- Hamed Chok, "Spatio-Temporal Association Rule Mining Framework for Estimating Missing Data in Sensor Networks and Analyzing Trend Evolution of Co-Evolving Multidimensional Data Streams," Master's Thesis, School of Computer Science, University of Oklahoma, March 2009.
- Hamed Chok and Le Gruenwald, "An Online Spatio-Temporal Association Rule Mining Framework for Analyzing and Estimating Sensor Data," in the proceeding of the International Database Engineering and Applications Symposium (IDEAS), September 2009.
- Hamed Chok and Le Gruenwald, "Spatio-Temporal Association Rule Mining Framework for Real-time Sensor Network Applications," in the proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), November 2009.

- Shiblee Sadik, Le Gruenwald, “Security for Data Stream Management Systems,” a chapter submitted to the book titled "Security in computing and networking systems: the state-of-the-art" edited by William McQuay and Walled W. Smari, November 2009.
- Le Gruenwald, Hanqing Yang, Md. Shiblee Sadik, and Rahul Shukla, "Using Data Mining to Handle Missing Data in Multi-Hop Sensor Network Applications, " in the proceedings of the 9th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE), June 2010.
- Md. Shiblee Sadik, “Outlier Detection for Data Streams,” Master’s Thesis, School of Computer Science, University of Oklahoma, July 2010.
- Le Gruenwald, Md. Shiblee Sadik, Rahul Shukla, and Hanqing Yang, "DEMS: A Data Mining Based Technique to Handle Missing Data in Mobile Sensor Network Applications, " In the proceedings of the 7th International Workshop on Data Management for Sensor Networks (DMSN), September 2010.
- Md. Shiblee Sadik and Le Gruenwald, "DBOD-DS: Distance Based Outlier Detection for Data Stream, " in the proceedings of 21st International Conference on Database and Expert Systems Applications (DEXA), September 2010.
- Md. Shiblee Sadid and Le Gruenwald, “An Adaptive Outlier Detection Technique for Data Streams,” submitted to International Conference on Statistical and Scientific Data Management, February 2011.
- Hamed Chok and Le Gruenwald, “Knowledge Discovery and Data Estimation in Sensor Network Databases,” under preparation for submission to the IEEE Transactions on Knowledge and Data Engineering, March 2011.
- Le Gruenwald, Md. Shiblee Sadik, Rahul Shukla, and Hanqing Yang, “A Data Mining Framework to Handle Missing Data for Wireless Sensor Networks,” under preparation for submission to the IEEE Transactions on Knowledge and Data Engineering, March 2011.
- Md. Shiblee Sadik and Le Gruenwald, “Outlier Detection for Data Streams,” under preparation for submission to the Journal of Very Large Data Bases, March 2011.

3. CONCLUSIONS

Through this project, we have successfully developed data mining based techniques to estimate values of missing sensor data for different types of sensor networks: centralized sensor network with single hop, centralized sensor network with multiple hops, distributed sensor network with multiple servers, and mobile sensor network. We have conducted experiments to compare the performance of our techniques with existing techniques using real life datasets obtained from both non-NASA and NASA applications as well as synthetic datasets. The NASA application datasets include those using micro-sensor networks provided from the NASA Sensor Webs project at JPL [NASA/JPL, 2010] as well as those using satellites, such as the DORIS dataset [DORIS] for space missions, Global Hydrology dataset [MSU] for global temperature monitoring, and the Surface Meteorology and Solar Energy Dataset [Data-Energy] generated by the NASA Goddard Earth Observing System. Through collaboration with Dr. Nikunj C. Oza at NASA Ames Research Center, we were also able to make use of the NASA spectral dataset for additional testing.

Our comprehensive experimental results show that overall our techniques provide the best accuracy in estimating the values of missing sensor data, while requires acceptable execution time for practical sensor network applications. In addition, our techniques are shown to save a significant amount of energy consumption compared with the approach in which missing data need to be retransmitted.

In terms of human resource contributions, this project provided financial support to five Master's students and four PhD students, three of whom are female. In terms of publications, the project has resulted in eighteen publications (fourteen already published, one submitted, and three under preparation).

4. REFERENCES

[Abadi, 2003] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, C. Erwin, E. Galvez, M. Hatoun, J. Hwang, A. Maskey, A. Rasin, A. Singer, M. Stonebraker, N. Tatbul, Y. Xing, R. Yan, S. Zdonik. "Aurora: A Data Stream Management System," In proceedings of the 2003 ACM SIGMOD International Conference on Management of data, San Diego, CA, 2003, Pages 666 – 666.

[Aggarwal, 1993] R. Aggarwal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," In Proceedings of the 1993 ACM SIGMOD Conference, 1993, Pages 207 – 216.

[Al-Karaki, 2004] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," IEEE Wireless Commun. Mag., vol. 11, no. 6, 2004, Pages. 6–28.

[Chok, 2009a] Hamed Chok and Le Gruenwald, "An Online Spatio-Temporal Association Rule Mining Framework for Analyzing and Estimating Sensor Data," In proceedings of the 2009 International Database Engineering & Applications Symposium, Cetraro - Calabria, Italy, 2009, Pages 217-226.

[Chok, 2009b] Hamed Chok and Le Gruenwald, "Spatio-temporal association rule mining framework for real-time sensor network applications," In proceeding of the 18th ACM conference on Information and knowledge management, Hong Kong, China, 2009, Pages 1761-1764.

[Clouqueur, 2002] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K.K Saluja, "Sensor Deployment Strategy for Target Detection," In proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, 2002, Pages 42-48.

[Dapple, 2004] UCL Carbon Monoxide Data Collection at Dapple, http://www.cs.ucl.ac.uk/research/vr/Projects/envesci/DAPPLE2004/UCLDAPPLE.html, Accessed May 2010.

[DORIS] http://cddis.gsfc.nasa.gov/doris_summary.html, last accessed 2010.

[DORIS-log] http://ids-doris.org/system/doris-system-events.html, last accessed 2010.

[Elderton, 1969] W. P. Elderton, N. L. Johnson. "Systems of Frequency Curves," Cambridge University Press, 1969.

[Energy-data] http://eosweb.larc.nasa.gov/sse/, last accessed 2010.

[Giannella, 2003] C. Giannella, J. Han, J. Pei, X. Yan, and P. Yu. in H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), "Mining Frequent Patterns in Data Streams at Multiple Time Granularities," In proceedings of the Next Generation Data Mining, AAAI/MIT, 2003.

[Gruenwald, 2007] Le Gruenwald, Hamed Chok, and Mazen Aboukhamis, "Using Data Mining to Estimate Missing Sensor Data," In proceedings of the Seventh IEEE International Conference on Data Mining Workshops, 2007, Pages 207-212.

[Gruenwald, 2010a] Le Gruenwald, Hanqing Yang, Md. Shiblee Sadik, and Rahul Shukla, "Using Data Mining to Handle Missing Data in Multi-Hop Sensor Network Applications," In proceedings of the 9th ACM International Workshop on Data Engineering for Wireless and Mobile Access, Indianapolis, Indiana USA, 2010.

[Gruenwald, 2010b] Le Gruenwald, Md. Shiblee Sadik, Rahul Shukla, and Hanqing Yang, "DEMS: A Data Mining Based Technique to Handle Missing Data in Mobile Sensor Network Applications," In proceedings of the 7th International Workshop on Data Management for Sensor Networks, Singapore, 2010.

[Halatchev, 2005] Mihail Halatchev, Le Gruenwald, "Estimating Missing Values in Related Sensor Data Streams," In proceedings of 11th International Conference on Management of Data, 2005, Pages 83 97.

[Heinzelman, 2000] W. R. Heinzelman, A. Chandrakasan,H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," In proceedings of 33rd Hawaii International Conference on System Sciences, IEEE , 2000 , Page10.

[Intel, 2009] Intel Berkeley Research Lab. http://db.csail.mit.edu/labdata/labdata.html, accessed 2009.

[Ibriq, 2004] J. Ibriq and I. Mahgoub, "Cluster-based Routing in Wireless Sensor Networks: Issues and Challenges," In proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2004, Pages. 759-766.

[Jiang, 2007] Nan Jiang and Le Gruenwald, "Estimating Missing Data in Data Streams," In proceedings of the 12th international conference on Database systems for advanced applications, Bangkok, Thailand, 2007, Pages 981-987.

[Jollife 2002] I.T. Jolliffe. “Principle Component Analysis”. Journal of the American Statistical Association, Vol 98, Issue January, Year 2003, Pages 1082 - 1083.

[Kay, 1993] S. Kay, “Fundamentals of Statistical Signal Processing: Estimation Theory,” Prentice Hall, 1993.

[Liu, 2005] B. Liu, Peter Brass, Olivier Dousse, Philippe Nain, and Don Towsley, “Mobility Improves Coverage of Sensor Networks,” In proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, Urbana-Champaign, IL, USA, 2005, Pages 300 – 308.

[Madden, 2005] S. Madden, M. Franklin, J. Hellerstein and W. Hong, “TinyDB: An Acquisitional Query Processing System for Sensor Networks,” ACM Transactions on Database Systems 2005, Volume 30, Issue 1, Pages 122 – 173.

[Mainwaring, 2002] A. Mainwaring, D. Culler , J. Polastre , R. Szewczyk , and J. Anderson, “Wireless Sensor Networks for Habitat Monitoring,” In proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA 2002, Pages 88-97.

[McKeeman, 1962] W. M. McKeeman, “Algorithm 145: Adaptive numerical integration by Simpson's rule,” Source Communications of the ACM, Vol. 5, 1962, Pages 604.

[McLachlan, 1997] G. McLachlan and T. Krishnan, “The EM algorithm and extensions,” Wiley series in probability and statistics, 1997.

[Metar, 2010] Metar, http://metar.noaa.gov/, Jan 2010.

[Meguerdichian, 2001] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M.B.Srivastava, “Coverage Problems in Wireless Ad-Hoc Sensor Networks,” In proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, 2001, Pages 1380-1387.

[Mhatre, 2004] Vivek Mhatre, Catherine Rosenberg. “Homogeneous Vs. heterogeneous sensor networks: A comparative study”, In the proceedings of IEEE International Conference on Communications, 2004, Pages 3646 – 3651.

[Motwani, 2003] R. Motwani, J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, G. Manku, C. Olston, J. Rosenstein, R. Varma. “Query Processing, Approximation, and Resource Management in a Data Stream Management System,” In proceedings of the Conference on Innovative Data Systems Research (CIDR), 2003, Pages 245-256.

[MSU] http://ghrc.nsstc.nasa.gov/uso/ds_docs/msu/msul90_dataset.html #p5, last accessed 2010.

[MSU-Desc] http://noaasis.noaa.gov/NOAASIS/ml/genlsatl.html, last accessed 2010.

[MSU-Data] ftp://ghrc.nsstc.nasa.gov/pub/data/msu, last accessed 2010.

[MSU-log] http://ghrc.nsstc.nasa.gov/uso/images/msu_sample.html, last accessed 2010.

[NASA/JPL, 2010] NASA/JPL Sensor Webs Project, http://caupanga.huntington.org/swim/, accessed August 2010.

[Ossama, 2004] Ossama Younis, Sonia Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", In the proceedings of IEEE transactions on Mobile Computing, 2004, Pages 366 – 379.

[Papadimitriou, 2005] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming Pattern Discovery in Multiple Time-Series," In proceedings of the 31st international conference of very large databases, 2005, Pages 697 - 708.

[Schwiebert, 2001] L. Schwiebert, S. Gupta, and J. Weinmann, "Research Challenges in Wireless Networks of Biomedical Sensors," In proceedings of the 7th annual international conference on Mobile computing and networking, Rome, Italy, 2001, Pages 151 – 165.

[Shafer, 1995] J. Shafer, "Model-Based Imputations of Census Short-Form Items," the Annual Research Conference, 1995

[Sibley, 2002] G.T. Sibley, M.H. Rahimi and G.S. Sukhatme, "Robomote – A tiny Mobile Robot Platform for Large-Scale Sensor Networks," In proceedings of IEEE International Conference on Robotics and Automation, 2002, Pages 1143-1148.

[Silberstein, 2006] A. Silberstein , R. Braynard , J. Yang, "Constraint Chaining: On Energy-Efficient Continuous Monitoring in Sensor Networks," In Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD '06), Chicago, Illinois, USA, June 2006. Pages 157-168.

[Silberstein, 2006] A. Silberstein, K. Munagala, and J. Yang, "Energy-Efficient Monitoring of Extreme Values in Sensor Networks," ACM SIGMOD, 2006, Pages 169 – 180.

[Srivastava, 2004] Ashok N Srivastava, Nikunj C. Oza and Julienne Stroeve, "Virtual Sensors: Using Data Mining Techniques to Efficiently Estimate Remote Sensing Spectra," IEEE Transaction on Geoscience and Remote Sensing.

[Szewczyk, 2004] R. Szewczyk, J. Polastre, A. M. Mainwaring, and D. E. Culler, "Lessons from a Sensor Network Expedition," In Proceedings of the First IEEE European Workshop on Wireless Sensor Networks and Applications, 2004, Pages 307-322.

[Tolle, 2005] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, W. Hong, "A macroscope in the redwoods," In proceedings of

the Third International Conference on Embedded Networked Sensor Systems (Sensys), San Diego, CA, 2005, Pages 51-63

[Vijayakumar, 2009] N. Vijayakumar and B. Plale, "Missing Event Prediction in Sensor Data Streams Using Kalman Filters," Knowledge Discovery from Sensor Data, Eds. Auroop R. Ganguly , João Gama , Olufemi A. Omitaomu , Mohamed Medhat Gaber and Ranga Raju Vatsavai, Published by CRC Press, 2009, Pages 149–170.

[Wang, 2005] G. Wang, G. Cao, T. parta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," In proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 2005, Pages 2302-2312.